

**Advantage™ VISION:Builder®**

**Advantage™ VISION:Two™ for  
OS/390®**

## **Installation Guide**

**14.0**



Computer Associates™

BUINM140.PDF/D92-010-014

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, the user may print a reasonable number of copies of this documentation for its own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software of the user will have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA).

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

## Chapter 1: Introduction

About this Book .....	1-1
Audience.....	1-1
VISION:Two Users.....	1-2
OS/390 SMP/E Facility.....	1-2
License Management Program.....	1-2
Installation Process.....	1-2
System Tape.....	1-3
Installation Overview .....	1-4
Licensing Requirements .....	1-4
Coding and Integrating Your Licensing Key .....	1-5
System Tape Unload .....	1-7
Installation Preparation Dialog .....	1-7
SMP/E Setup and the Basic Installation .....	1-8
Customizing and Setups .....	1-9
CD-ROM Contents .....	1-9
About the Online Documentation .....	1-9
Installing Online Documentation and the Acrobat Reader .....	1-9
Viewing Online Documentation .....	1-10
Using Adobe Acrobat Reader.....	1-10
Contacting Total License Care (TLC).....	1-10
Contacting Computer Associates.....	1-11

## Chapter 2: System Tape Unload

Step 1 – Copy System Tape File 1 to a PDS.....	2-1
Step 2 – Copy System Tape Files 2 through 16 to Disk Data Sets.....	2-2

---

## Chapter 3: Installation Preparation Dialog

Tips and Hints about Using the Installation Preparation Dialog.....	3-2
Dialog Navigation .....	3-2
Basic SMP/E Concepts .....	3-4
Step 3 – Complete the IP Dialog.....	3-5
Initialization Display .....	3-5
Panel Display .....	3-9
Variables Panels .....	3-10
JCL Panels .....	3-19

## Chapter 4: SMP/E Setup and Basic Installation

Step 4 – Allocate Data Set.....	4-2
Step 5 – Define the CSI and the Global, Distribution, and Target Zones .....	4-3
Step 6 – RECEIVE the MCS and SYSMODS into the Global Zone .....	4-3
Step 7 – RECEIVE the PTF and APAR SYSMODS into the Global Zone .....	4-4
Step 8 – APPLY the VISION:Builder Elements (SYSMODS) to the Target Libraries.....	4-4
Step 9 – APPLY the VISION:Builder PTF SYSMODS to the Target Libraries.....	4-4
Step 10 – Run the Installation Verification Procedure using the Target Load Library. ....	4-5
Step 11 – ACCEPT the VISION:Builder Elements (SYSMODS) to the Distribution Libraries .....	4-6
Step 12 – ACCEPT the VISION:Builder PTF SYSMODS to the Distribution Libraries .....	4-6

## Chapter 5: Customizations and Setups

Step 13 – APPLY Customizing APARs.....	5-1
Customization Activity Considerations .....	5-4
Copying the VISION:Builder System Load Library .....	5-5
Step 14 – Customize the Parameter Modules.....	5-5
Modifying the Modules.....	5-6
Storing the Modules.....	5-6
Step 15 – Install the DB2 Database Access Module MARKSQL.....	5-6
Using Slots to Access DB2 Tables.....	5-6
Controlling the Number of Statement Slots .....	5-7
Assembling and Preparing MARKSQL .....	5-7
Using the BIND Function.....	5-8
Teradata Information .....	5-11
Step 16 – Install the PAL File Definitions and Requests .....	5-11
Step 17 – Relink Static Own Code Integration .....	5-12
Error Messages to Ignore .....	5-12

---

Step 18 – Set Up for Use with the TSO Command Processor .....	5-13
OQL and BQL Parameter Module Modification .....	5-13
TSO Help Data Set .....	5-14
OLX Command Processors .....	5-14
Step 19 – Copy VISION:Builder Message Modules to LPA .....	5-14
Step 20 – Install VISION:Workbench for DOS .....	5-15
Step 21 – Set Up VISION:Workbench for ISPF Requirements .....	5-15
Allocating VISION:Workbench for ISPF Run-Time Libraries .....	5-16
Allocation Requirements .....	5-16
Panel Library .....	5-17
Locate Text .....	5-17
Locate Skeletons .....	5-18
Generate Facility .....	5-18
IMPORT Option .....	5-18
Start-up CLIST .....	5-18
Using STEPLIB and System Link Library .....	5-19
Using the LIBDEF Feature .....	5-19
More about ISPF FILE Allocations .....	5-19
List Data Set and Internal Work Files .....	5-20
The List Data Set .....	5-20
Internal Work Files .....	5-21
Invoking VISION:Workbench for ISPF .....	5-22
Adding an Option to a Standard ISPF Primary Menu .....	5-23
Using the ISPSTART Command .....	5-25
Using Other VISION:Workbench for ISPF Subsections .....	5-26
Optional Setup for VISION:Workbench for ISPF .....	5-27
Preprocessing Your Panel Library .....	5-27
Customizing Job Submission Skeletons .....	5-29
LMF SUPPORT .....	5-30
Step 22 – Quick Start Utility Setup .....	5-30

## Chapter 6: Maintenance and Support

Maintenance – Installing the PTFs and APARs .....	6-2
Support – Problem Reporting .....	6-3
VISION:Builder and COMLIB Problem Reporting .....	6-3
VISION:Workbench for DOS Problem Reporting .....	6-4
VISION:Workbench for ISPF Problem Reporting .....	6-4
Panel Identification .....	6-4
Unexpected Error Panel .....	6-4
User Code .....	6-5

---

## Appendix A: JCL

## Appendix B: VISION:Builder Parameter Modules

M4PARAMS and M4LEPARM .....	B-1
M4SFARM .....	B-17
Defining Additional Data Validation Symbols .....	B-17
Change Automatic Date Validation Format .....	B-18
Example of Additional Source Statements .....	B-20
MARKLIBP .....	B-21
MARKSQL .....	B-23
Query Language Parameters – BQLPARM .....	B-30
Online Language Parameters – OQLPARM .....	B-37

## Appendix C: Sample ISPF Startup CLIST

## Appendix D: Invocation Panels

XSR@PRIM .....	D-1
M9PRIM .....	D-2

## Appendix E: Skeleton and User Panel Listings

## Index

# Introduction

---

Thank you for choosing VISION:Builder 14.0 or VISION:Two 14.0. Before you install the software, read this chapter for important information.

This book describes how to install VISION:Builder® and VISION:Two™. Any questions regarding the installation should be directed to Computer Associates® Technical Support. For more information, see [Contacting Computer Associates on page 1-11](#).

## About this Book

Read this chapter to acquire an understanding of the elements and processes that comprise the installation of VISION:Builder at your site.

Read the subsequent chapters in this book before starting the installation process so that you can have all of the details regarding the installation, customization, and maintenance of VISION:Builder.

## Audience

The System Programming Group is usually responsible for software product installation and maintenance because of their SMP/E (System Modification Program Extended) knowledge. This book assumes a working knowledge of the SMP/E Facility and its processes.

A basic standalone SMP/E Install and Maintenance approach is presented. For the knowledgeable SMP/E User, there is enough information provided in this book, and the generated JCL and Control Statements, to allow integration with any site specific SMP/E standards. For the SMP/E Novice, this book and the Dialog HELP Facility should provide enough of the basic information and concepts you need to complete the basic SMP/E installation process.

## VISION:Two Users

If you are a VISION:Two user, whenever the term VISION:Builder is used in the book, it also refers to VISION:Two. There are notations within this book to indicate whether something is specific to VISION:Builder, VISION:Two, or a specific Database Option.

## OS/390 SMP/E Facility

Starting with release 14.0 of VISION:Builder and VISION:Two, the Installation and Maintenance is managed by and under the control of the OS/390<sup>®</sup> SMP/E facility as provided by IBM. This process differs significantly from previous releases (13.8 and prior) of VISION:Builder and VISION:Two.

## License Management Program

VISION:Builder uses the Computer Associates License Management Program (LMP), which provides a standardized and automated approach to the tracking of licensed software.

## Installation Process

VISION:Builder is delivered on a Tape Cartridge. An LMP Product key certificate contains your execution key for each CPU licensed at your site. Other identifying information is provided on the external tape cartridge label.

Save all output generated during the installation, along with the system tape, for future reference.

The basic SMP/E setup and installation process is identical for all users. The first file on the system tape contains JCL for a job that transfers all the remaining system tape files to disk data sets. Once the system tape files are transferred to disk, you will have all of the elements that you need to prepare and complete the installation, customizing, and maintenance processes. Use a simple interactive ISPF dialog invoked by a REXX Exec to tailor the JCLs.

## System Tape

The VISION:Builder system tape supplied for the OS/390 environment is a standard labeled tape cartridge containing 16 files. The following table shows the order and content of the nine files on the tape.

File	Description
1	A WORK.PDS that contains the JCL (BLCOPY2) for a job that will transfer all the system tape files to disk data sets.
2-6	The data sets needed to run the Installation Preparation Dialog under ISPF.  The “PREP” data sets contain all the elements used by the VISION:Builder Installation Preparation Dialog to tailor and build all the install, customizing and maintenance JCLs and Control Statements.
2	The PREP.CLIST file contains REXX Execs used under ISPF to run the Dialog.
3	The PREP.PANELS file contains ISPF Dialog Panels and Helps.
4	The PREP.MSGS file contains Dialog Messages.
5	The “PREP.SKELS” has all the JCL and Control Statement models that need “tailoring”. These will be “File Tailored” by the Dialog into the PREP.JCLCNTL data set.
6	The PREP.JCLCNTL file contains JCL, Control Statements, PTFs (Program Temporary Fixes), and APARs (Authorized Program Analysis Reports) used during the installation, customization, and maintenance processes that DO NOT need tailoring.  <b>Note:</b> The PDS Data Set (...PREP.JCLCNTL) is populated with tailored JCL and Control Statements as described in Chapter 3 Install Prep Dialog. This data set is referenced in this book as the PDS data set (...PREP.JCLCNTL).
7-15	The VISION:Builder data sets. These will be the “Indirect” data sets that will be used by the SMP/E Facility to build the target and distribution libraries.
7	The SMPE.I.BLOAD file contains the VISION:Builder engine load modules.
8	The SMPE.I.BLSAMP file contains members (source code, control statements, etc.) used to customize VISION:Builder.
9	The SMPE.I.CLLOAD file contains the COMLIB Component Load Modules.
10	The SMPE.I.WBLOAD file contains the VISION:Workbench for ISPF Load Modules.

File	Description
11	The SMPE.I.WBCLIST file contains the VISION:Workbench for ISPF CLIST Members.
12	The SMPE.I.WBPANEL file contains the VISION:Workbench for ISPF Panel Members.
13	The SMPE.I.WBMSGGS file contains the VISION:Workbench for ISPF Messages Members.
14	The SMPE.I.WBSKELS file contains the VISION:Workbench for ISPF Skeleton Members.
15	The SMPE.I. SCLINK file contains the SAS/C Link Library (Runtime) Load Modules.
16	The USER.SAMPLIB file contains miscellaneous samples for User Reference.

## Installation Overview

The VISION:Builder installation is divided into the following sections:

- [Licensing Requirements on page 1-4](#)
- [Coding and Integrating Your Licensing Key on page 1-5](#)
- [System Tape Unload on page 1-7](#)
- [Installation Preparation Dialog on page 1-7](#)
- [SMP/E Setup and the Basic Installation on page 1-8](#)
- [Customizing and Setups on page 1-9](#)

## Licensing Requirements

VISION:Builder interfaces with the Computer Associates Licensing System using the CA TNG Framework for OS/390 Common Services CAIRIM and its CA-LMP facility, which is used to track licensed software.

For more information regarding the CA TNG Framework for OS/390 Common Service CAIRIM and its CA-LMP facility, refer to the *Unicenter TNG Framework for OS/390 Reference Guide* and the *Unicenter TNG Framework for OS/390 Installation and Maintenance Guide*.

CA-LMP (License Management Program) is a standardized and automated approach for tracking licensed software. CA-LMP is provided as an integral part of CAIRIM, and is required for VISION:Builder to initialize properly.

If CAIRIM has not already been installed on your system, you must install it before you install and use VISION:Builder Release 14.0. Refer to the Unicenter TNG Framework for OS/390 documentation for information about installing CAIRIM.

**Note:** Once CAIRIM has been installed or maintained at GenLevel 9212 or above, CA-LMP support will be available for all Computer Associates products that support CA-LMP.

## Coding and Integrating Your Licensing Key

The first task for the Installation of VISION:Builder is to get your Computer Associates Licensing Key information coded and integrated into the CAIRIM CA-LMP facility. This is a standard function for all Computer Associates software products. You must add a record with your VISION:Builder CA-LMP Execution Key information, as provided on the key certificate, to the the KEYS member in the CAIRIM parameter data set, at the OPTLIB DD statement.

The CA-LMP key certificate you received with VISION:Builder contains the following information:

Field	Description
Product Name	The trademarked or registered product name as licensed for the designated site and the CPUs.
Product Code	A two-character code for the VISION:Builder System and two-character codes for each of the licensed Database Options.
Supplement	The reference number of your license for VISION:Builder which may be in the format nnnnnnn - nnn.
CPU ID	The code identifying the specific CPU on which VISION:Builder is to be installed.
Execution Key	An encrypted code required by CA-LMP for VISION:Builder initialization. This is also referred to as the LMP Key.
Expiration Date	The date (ddmmmyy) your license for VISION:Builder expires.
Technical Contact	The name of the technical contact at your site who is responsible for the installation and maintenance of this licensed copy of VISION:Builder. This is the person to whom Computer Associates addresses all CA-LMP correspondence.
MIS Director	The name of the Director of MIS (or the person who performs this function at your site). If a person's name is omitted from the certificate, you should supply the actual certificate when correcting and verifying it.
CPU Location	The address of the building containing the CPU on which VISION:Builder is installed.

You must add the CA-LMP execution key information, as provided on the key certificate, to the CAIRIM parameters to ensure that VISION:Builder initializes properly. To define a CA-LMP execution key to the CAIRIM parameters, modify the KEYS member in the OPTLIB data set.

The parameter structure for member KEYS is:

PROD(pp) DATE(ddmmmyy) CPU(tttt-mmmm/ssssss) LMPCODE(kkkkkkkkkkkkkk)

pp	The two-character product code for VISION:Builder; required.
SM	CA-VISION:Builder Engine and Components
SZ	CA-VISION:Builder Generalized Data Base Interface (GDBI)
S4	CA-VISION:Builder IMS Data Base Option
S6	CA-VISION:Builder DB2 Data Base Option
S8	CA-VISION:Two Engine and Components
S9	CA-VISION:Two Generalized Data Base Interface (GDBI)
TF	CA-VISION:Two IMS Data Base Option
TG	CA-VISION:Two DB2 Data Base Option
ddmmmyy	The CA-LMP licensing agreement expiration date (for example, 15JAN02).
tttt-mmmm	The CPU type and model (for example, 3090-0600) on which the product is to run; required. If the CPU type and/or model are less than four characters, insert blank spaces for the unused characters.
ssssss	The serial number of the CPU on which the product is to run; required.
kkkkkkkkkkkkkk	The execution key needed to run the product; required. The CA-LMP execution key can be found on the key certificate that was shipped with the product.

Here is an example of the parameter entry for the CA-LMP:

PROD(SM) DATE(31JAN03) CPU(3090-0600/315109) LMPCODE(5149K01131R08ES)

For more information about defining the CA-LMP execution keys using the CAIRIM parameters, refer to the *Unicenter TNG Framework for OS/390 Installation and Maintenance Guide*.

## System Tape Unload

File 1 on the system tape contains the JCL (BLCOPY2) for a job that transfers all the system tape files to disk data sets. The only JCL that needs to be prepared by the installer is an IEBCOPY JCL to copy the contents of file 1 to a PDS. This JCL is shown in [Figure 2-1 on page 2-1](#) as part of the first step of the installation process.

Once the first system tape file is unloaded, you review, tailor, and submit the JCL in member BLCOPY2. This transfers system tape files 2-16 to disk data sets. The main consideration here is the High-level Index Name assigned to the disk data sets.

[Appendix A, JCL](#) contains an alphabetic list of all the JCL members referenced in the Installation procedures. A detail list of the delivered JCL members follows the alphabetic list.

Once all the system tape files are copied to disk data sets, all the elements (JCL, control statements, source, load modules, etc.) needed for the VISION:Builder Installation process are available to the installer. At this point, you can save the system tape; you only need the disk data sets to complete the installation process.

See [Chapter 2, System Tape Unload](#) for more information.

## Installation Preparation Dialog

In order to generate and build all the JCL and Control Statements needed during the installation, customization, and maintenance processes of VISION:Builder, an Interactive ISPF Installation Preparation dialog has been provided as part of VISION:Builder. The ISPF Dialog is controlled by a REXX Exec that is started from TSO/ISPF Option 6.

The Displays within the Dialog will prompt you for the various values needed to generate and tailor all the JCL and Control Statements referenced in this book. The data sets containing the REXX Exec and ISPF Dialog elements are among the data sets that are unloaded from the VISION:Builder System Tape. There are no special requirements needed for your ISPF system in order to run the dialog. Your TSO Logon Region Size should be at least 4096.

All values entered during the Dialog Session(s) are saved in your Profile Variables and returned to the displays on subsequent Dialog invocations. The Dialog can be terminated and restarted at any time. The generated JCLs and Control Statements are placed into a PDS for subsequent use during the Installation, Customizing and Maintenance procedure detailed in this book.

See [Chapter 3, Installation Preparation Dialog](#) for more information.

## SMP/E Setup and the Basic Installation

This portion of the installation process uses some of the JCL and control statements prepared by the Installation Preparation Dialog. There are jobs that define and allocate the following items:

- SMP/E target and distribution libraries for VISION:Builder
- A SMP/E CSI (Consolidated Software Inventory) library for tracking the activities
- SMP/E work data sets

The SMP/E setup has the following steps:

1. The elements from the Indirect libraries, unloaded from the VISION:Builder system tape, are stored into the SMP/E work data sets using the RECEIVE operation. These elements include all the latest PTFs and APARs for VISION:Builder.
2. The SMP/E APPLY processing is performed to update the target libraries based on the Modification Control Statements (MCS) and a SMPJCLIN job stream. Essentially, the elements from the SMP/E work data sets are transferred to the target libraries. These elements include all the latest PTFs.

**Note:** The APARs are special items and are handled during the Customizing and Setups portion of the installation process.

3. The target load library containing VISION:Builder is used in a job stream that performs an installation verification process. This confirms that VISION:Builder was installed and prepared correctly.
4. The SMP/E ACCEPT processing is performed to update the Distribution Libraries based on the Modification Control Statements (MCS) and a SMPJCLIN job stream. Basically, the elements from the SMP/E work data sets are transferred to the Distribution Libraries. This includes all the latest PTFs.

**Note:** The APARs are special items and are handled during the Customizing and Setups portion of the installation process.

At this point, the SMP/E Setup and the Basic Installation are complete. The target and distribution libraries are synchronized. The SMP/E concept is to APPLY to the target and test the update/PTF/APAR. If you are not satisfied with the tests, you can RESTORE the modified target elements to their previous state from the distribution libraries. If the modifications perform as expected, you permanently ACCEPT the modifications into your distribution libraries. There is no direct method for undoing modifications once the ACCEPT processing is run.

See [Chapter 4, SMP/E Setup and Basic Installation](#) for more information.

## Customizing and Setups

With a software system as diverse as VISION:Builder, there are several capabilities and facilities that may need some additional setup to operate within each installation site environment. The setup steps that apply to each site vary depending on how the software system and components are used at each site. Also, some portions of the software system and its components can be customized to meet the specific needs and requirements of your site.

Customizing and Setups can be performed at any time and are optional. As your specific needs and requirements change, you may want to make additional adjustments.

See [Chapter 5, Customizations and Setups](#) for more information.

## CD-ROM Contents

- Online documentation
- Adobe® Acrobat® Reader software and Acrobat Help

### About the Online Documentation

The CD-ROM contains the documentation for VISION:Builder. The documents, called books, are in Adobe Acrobat Portable Document Format (PDF) and are designed for you to read online using the Acrobat Reader.

Each online document contains a table of contents, index, and cross-references.

**Note:** You can install the online documentation only on a Windows® system.

### Installing Online Documentation and the Acrobat Reader

You can install the online documentation on your local hard drive or on a network server. Alternately, you can access the documentation directly from the CD-ROM.

If you do not have Acrobat Reader installed, you can install it from the CD-ROM.

To install the online documentation, the Acrobat Reader, or both:

1. Close all application programs.
2. Insert the CD-ROM into the CD-ROM drive.
3. Click the Start menu and select Run.
4. In the Run dialog box, type: D:\Books\Setup.exe (where D:\ is the CD-ROM drive) and click OK.

5. Follow the instructions. Computer Associates recommends that you install the online documentation in the default directory (C:\ProgramFiles\CA\Advantage VISION\_Builder VISION\_Two 14.0 OS390\Books\) or a directory of your choice (for example, C:\Advantage VISION\_Builder VISION\_Two 14.0\Books\).

### Viewing Online Documentation

Regardless of the location of the online documentation (on a local drive, a network server, or CD-ROM), you can view the online documentation using the following methods:

- In Windows, click the Start menu, point to Programs, point to Advantage VISION\_Builder VISION\_Two 14.0 OS. Double-click the PDF file name.
- In Windows Explorer, point to the Books directory on the hard drive where you installed the online documentation. Double-click the PDF file name.
- In Windows Explorer, point to the Books directory on the CD-ROM drive and double-click the PDF file name.

### Using Adobe Acrobat Reader

Use Acrobat Reader to view the online documentation, adjust the size of the page, and perform searches. For more information, use the Acrobat Help menu.

## Contacting Total License Care (TLC)

TLC is available Monday-Friday 7 am - 9 pm Eastern Time in North America and 7 am - 7 pm United Kingdom time. Additionally, 24-hour callback service is available for after hours support. Contact TLC for all your licensing requirements.

Be prepared to provide your site ID for product activation.

To activate your product, use one of the following:

Location	Phone	email
<b>North America:</b>	800-338-6720 (toll free) 631-342-5069	help@licensedesk.cai.com
<b>Europe:</b>	00800-1050-1050	euro.tlc@ca.com
If your company or local phone service does not provide international access, please call your local Computer Associates office and have them route you to the above number.		

Location	Phone	email
Australia:	1-800-224-852	
New Zealand:	0-800-224-852	
Asia Pacific:	800-224-852	
Brazil:	55-11-5503-6100	
Japan:	Not available	JPNTLC@ca.com

## Contacting Computer Associates

For technical assistance with this product, contact Computer Associates Technical Support on the Internet at [esupport.ca.com](mailto:esupport.ca.com). Technical support is available 24 hours a day, 7 days a week.



# System Tape Unload

In this portion of the VISION:Builder installation, you copy the elements and components from the system tape to disk data sets. The following two steps are performed to create a group of disk data sets that contain the entire host portion of the VISION:Builder software system and components:

- [Step 1 – Copy System Tape File 1 to a PDS on page 2-1](#)
- [Step 2 – Copy System Tape Files 2 through 16 to Disk Data Sets on page 2-2](#)

Everything you need to prepare and set up the system is contained in the disk data sets.

## Step 1 – Copy System Tape File 1 to a PDS

In step 1, you copy the contents of file 1 on the VISION:Builder system tape from the tape to a PDS. The PDS contains a member named BLCOPY2. This member contains a JCL Job stream that will unload the remaining system tape files.

[Figure 2-1](#) shows the JCL to transfer file 1 to a PDS. Prepare the IEBCOPY job as described in [Figure 2-1](#).

```

/* MEMBER BLCOPY1
/******
/*
/* THIS JOB COPIES FILE 1
/* FROM THE VISION:UILDER SYSTEM TAPE TO DISK
/*
/* BEFORE YOU RUN THIS JOB, REVIEW JCL AND SPECIFY:
/*
/* THE INPUT TAPE INFORMATION: UNIT, VOLUME SERIAL NUMBER
/* THE OUTPUT DISK DATASET NAME, UNIT AND VOLUME SERIAL NUMBER.
/*
/******
/*
//COPY1 EXEC PGM=IEBCOPY,REGION=2M
//SYSPRINT DD SYSOUT=*
/*
//INPUT DD DSN=VISION.BUILDER.FILE1,
// DISP=OLD,
// LABEL=(1,SL,EXPDT=98000),
// UNIT=CART,
// VOL=(PRIVATE,RETAIN,SER=(TAPVOL))

```

Figure 2-1 Copy System Tape File 1 to a WORK.PDS Data Set (Page 1 of 2)

```
//*  
//OUTPUT DD DSN=BUILDER.R140.WORK.PDS,  
// DISP=(NEW,CATLG),  
// UNIT=SYSDA,  
// VOL=SER=DSKVOL,  
// SPACE=(TRK,(3,1,3)),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)  
//*  
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,15)  
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,15)  
//SYSIN DD *  
COPY INDD=INPUT,OUTDD=OUTPUT  
/*
```

Figure 2-1 Copy System Tape File 1 to a WORK.PDS Data Set (Page 2 of 2)

The names assigned to the disk data sets unloaded from the system tape are default names. The table on page [2-3](#) shows all the data set names and their characteristics. For continuity purposes, the names shown here are used throughout the installation process instructions and in all JCL examples.

Take the time to determine the high-level index to use for VISION:Builder data sets. Keep in mind that nine of the delivered data sets are the indirect libraries used by SMP/E when creating the target and distribution libraries. Take a moment to look ahead at Step 2 and the table listing the System Tape Files that will be unloaded to disk data sets.

## Step 2 – Copy System Tape Files 2 through 16 to Disk Data Sets

In step 2, you transfer all the remaining VISION:Builder system tape files to disk data sets. These disk data sets created in step 2, along with the WORK.PDS created in step 1, comprise the entire VISION:Builder software system and components.

Using the JCL member BLCOPY2 in the WORK.PDS, run a job to copy files 2 through 16 from the system tape to disk. The following table shows the list of all the data sets that are created by the system tape unload steps 1 and 2. The characteristics are shown along with basic space requirements. See [Appendix A, JCL](#) for the BLCOPY2 (and BLCOPY1) JCL that contains the detail characteristics for each data set.

The high-level index of **BUILDER.R140.** can be changed in the BLCOPY2 JCL using the in-stream procedure symbolic DSNHLQ. This high-level index is requested during the initial startup of the ISPF Installation Preparation Dialog described in [Chapter 3, Installation Preparation Dialog](#).

**Note:** The disk data set names assigned to the unloaded system tape files, and shown in this table, are default names. For continuity purposes, the names shown here are used throughout the installation process instructions and in all the JCL examples.

<b>Tape File #</b>	<b>Disk Data Set Name</b>	<b>DS ORG</b>	<b>REC FM</b>	<b>REC SIZE</b>	<b>BLK SIZE</b>	<b>TRKS</b>	<b>DIR BLKS</b>
1	BUILDER.R140.WORK.PDS	PO	FB	80	27920	3	1
2	BUILDER.R140.PREP.CLIST	PO	FB	80	27920	5	5
3	BUILDER.R140.PREP.PANELS	PO	FB	80	27920	5	10
4	BUILDER.R140.PREP.MSGS	PO	FB	80	27920	3	3
5	BUILDER.R140.PREP.SKELS	PO	FB	80	27920	15	15
6	BUILDER.R140.PREP.JCLCNTL	PO	FB	80	27920	30	20
7	BUILDER.R140.SMPE.I.BLLOAD	PO	U	0	32760	90	50
8	BUILDER.R140.SMPE.I.BLSAMP	PO	FB	80	27920	30	10
9	BUILDER.R140.SMPE.I.CLLOAD	PO	U	0	32760	50	15
10	BUILDER.R140.SMPE.I.WBLOAD	PO	U	0	32760	50	40
11	BUILDER.R140.SMPE.I.WBCLIST	PO	FB	80	27920	20	10
12	BUILDER.R140.SMPE.I.WBPANEL	PO	FB	80	27920	120	250
13	BUILDER.R140.SMPE.I.WBMSGs	PO	FB	80	27920	15	40
14	BUILDER.R140.SMPE.I.WBSKELS	PO	FB	80	27920	5	5
15	BUILDER.R140.SMPE.I.SCLINK	PO	U	0	32760	90	30
16	BUILDER.R140.USER.EXAMPLES	PO	FB	80	27920	20	10

The BLKSIZES of 27920 are coded as BLKSIZE=0 in the delivered BLCOPY2 JCL, whereas BLKSIZES of 32760 are coded as such.

The data set for file 6 is referenced throughout the installation, customization, and maintenance processes described in this book. It contains the JCL jobs and control statements. The reference is stated as follows:

```
the PDS Data Set (...PREP.JCLCNTL).
```

The data set for file 8 is referenced several times in this book. It contains the source code used during the customizing and setup steps. The reference is stated as follows:

```
the Samples Data Set (...BLSAMP).
```

After you perform step 2, the system tape unload is complete. Save the system tape as a backup.



## Installation Preparation Dialog

---

Now that all the system tape files have been unloaded to disk data sets, you need to prepare all the JCL and control statements used during the SMP/E installation, the customization, and the maintenance processes of VISION:Builder and its components.

To simplify the tailoring of your JCL and control statements, VISION:Builder contains a TSO/ISPF dialog, known as the Installation Preparation Dialog (sometimes referred to as the IP Dialog in this book), to guide you through this preparation process. All the data sets needed by the IP Dialog were unloaded from the system tape in Step 2. There are no special requirements needed for your ISPF system in order to run the Installation Preparation Dialog. The IP Dialog is started from TSO Option 6 using a REXX exec. Your TSO logon region size should be at least 4096.

The displays within the IP Dialog prompt you for the various values needed to generate and tailor all the JCL and control statements referenced in this book.

All values entered during the IP Dialog sessions are saved in your profile variables and returned to the displays on subsequent dialog invocations. The IP Dialog can be terminated and restarted at any time. The generated JCLs and control statements are placed into the PDS data set (...PREP.JCLCNTL) for subsequent use during the Installation, Customizing and Maintenance procedure detailed in this book.

**Note:** The IP Dialog contains a HELP facility and detailed descriptions of each panel and field value presented during your session. The same information is provided here for easy reference.

## Tips and Hints about Using the Installation Preparation Dialog

- Initially, during the first invocation of the IP Dialog, all the values that are displayed are the default values.
- Once the values are accepted or changed, those values that are entered become the values displayed in subsequent invocations of the IP Dialog.
- The IP Dialog can be started and stopped at any time. All entered and saved information will be available in the next session. The information is saved in your profile variables.
- You can erase all the profile variables created by the IP Dialog from your profile pool. This causes the next invocation of the IP Dialog to display the default values again.

To ERASE all profile variables created by this Dialog, enter the following execute command on the TSO/ISPF Option 6 screen:

```
EXEC 'BUILDER.R140.PREP.CLIST (DELVAR)'
```

'BUILDER.R140.PREP.CLIST(DELVAR)' is the data set and member containing the REXX Exec.

**Note:** The Data Set Name must match the name assigned the file that was unloaded from the VISION:Builder System Tape in Step 2.

- This Dialog *only prepares* the JCLs and control statements. The instructions for using and submitting the JCLs are described in [Chapter 4, SMP/E Setup and Basic Installation](#), [Chapter 5, Customizations and Setups](#), and [Chapter 6, Maintenance and Support](#).
- This Dialog assumes a working knowledge of the SMP/E facility and its processes. The HELP facility provides enough basic information to guide the novice through the process.

### Dialog Navigation

The Installation Preparation Dialog moves forward, panel to panel, gathering the necessary information, and then allowing you to verify (view) what you entered. You press Enter to advance from panel to panel.

When you need to change information or go back to a previous screen, use the END (PF3/F3) command. The information you have already entered and the values you have just keyed at an input panel are saved. When you get back to the Variables Part 1 Panel, the END (PF3/F3) command EXITS the IP Dialog. Remember that all your saved information will be redisplayed the next time you start the IP Dialog.

You use the CANCEL command to immediately EXIT the IP Dialog. When you are at an input panel, any information that you just entered is not saved. When you are at an Information Display panel, the entered information was already saved, so the CANCEL command just exits the IP Dialog.

When you EXIT the IP Dialog using the END or CANCEL commands, one of the following screens displays before you return to TSO/ISPF Option 6.

```
*****
***                                     ***
***           The Dialog was "ENDED" by User           ***
***                                     ***
*****

***
```

```
*****
***                                     ***
***           The Dialog was "CANCELLED" by User           ***
***                                     ***
*****

***
```

When you use the END (PF3/F3) command after keying values on an input panel, the IP Dialog first edits any values that you just keyed, performs the saves, and then exits or goes back to a previous screen. If an error is detected on a keyed value, an error message displays and you must correct the value before you END the IP Dialog. You could use the CANCEL command at this point to exit without invoking the edit and save process, but the values that you just keyed are not saved.

The following is an example of an input panel with an error message.

```
BLVAR1P ----- VISION:Builder Installation Preparation Dialog -----
OPTION ==>
The CSI Data Set Name must end with '.CSI' .
VARIABLES - Part 1 - SMP/E CSI Information

The OS/390 SMP/E Facility is designed to manage the Installation of
Software Products and track any modifications. SMP/E uses the CSI
(Consolidated Software Inventory) to keep records about the software.

Please provide the SMP/E CSI and SMPPTS Information: (Use HELP for details)

Will the CSI be New ?      => YES          Enter YES or NO
CSI Data Set Name (...CSI) => BUILDER.R140.CIS
CSI VOLUME ("NEW" CSI)    =>              (If Blank, No VOLUME Used)
SMPPTS - High-Level Index => BUILDER.R140
SMPPTS UNIT ("NEW" CSI)   => ISPDA
SMPPTS VOLSER ("NEW" CSI) =>              (If Blank, No VOLSER Used)
SMP/E Default Unit        => ISPDA
SMP/E Default Volser      =>              (If Blank, No VOLSER Used)

Use ENTER to Process the Information
Use END   to Save the Information and Exit the Dialog
Use CANCEL to Exit the Dialog Without Saving the entered Information
```

## Basic SMP/E Concepts

SMP/E uses the following zones to organize and structure the information and elements of a software system:

- The global zone contains information regarding the elements staged in the SMP/E work data sets, and the indexes to the related distribution and target zones.
- The distribution and target zones contain information about the elements in the distribution and target libraries.

The software system elements (SYSMODS) are RECEIVED and the information is recorded into the global zone and staged in the work or the indirect data sets.

An APPLY run places elements into the target libraries using information from the target zone and recording the activity in the global zone. The target library elements are exercised and tested by the user to make sure they are satisfied with the performance and stability of the SYSMOD elements.

If everything works as expected, an ACCEPT run places elements into the distribution libraries using information from the distribution zone and recording the activity in the global zone.

**Note:** Once an element (SYSMOD, PTF, or APAR) is ACCEPTED into the Distribution Libraries, you cannot RESTORE items back to their previous state or level.

Any future modifications (PTF and APAR SYSMODS) to the software system will follow the same basic flow.

There are points within this flow where Elements (SYSMODS) can be REJECTED from the global zone or RESTORED from the distribution libraries to the target libraries. See [Chapter 4, SMP/E Setup and Basic Installation](#) and [Chapter 5, Customizations and Setups](#) for more information.

## Step 3 – Complete the IP Dialog

A REXX exec controls the TSO/ISPF Installation Preparation Dialog. The IP Dialog is started from the TSO/ISPF Option 6 screen.

The IP Dialog contains the following sections:

- [Initialization Display](#) Use the panels in this section to get the high-level index name needed to access the unloaded system tape files. The high-level index name is also used to tailor the JCL and control statements.
- [Panel Display](#) Use the panels in this section to review and change all the variables needed for tailoring the JCL and control statements used during the installation, customization, and maintenance tasks associated with VISION:Builder.

The unloaded system tape data sets used by the IP Dialog are the ...PREP.CLIST, PANELS, MSGS, SKELS, and JCLCNTL. See [Step 2 – Copy System Tape Files 2 through 16 to Disk Data Sets on page 2-2](#) for the list of system tape data sets unloaded to disk.

### Initialization Display

At the Command Prompt on the TSO/ISPF Option 6 screen, enter an execute command, data set, member name, and optionally, an input parameter in the following format:

```
EXEC 'BUILDER.R140.PREP.CLIST(PREPINST)' 'BUILDER.R140'
```

'BUILDER.R140.PREP.CLIST(PREPINST)' is the data set and member containing the REXX exec.

**Note:** The data set name must match the name assigned the file that was unloaded from the VISION:Builder system tape in Step 2.

'BUILDER.R140' is the Option Input Parameter.

This should be the high-level index assigned to the unloaded system tape data sets.

The input parameter represents the high-level index name assigned to the unloaded system tape data sets in Step 2. If the default was used (BUILDER.R140) in Step 2, this parameter is not needed. You are given the opportunity to change this within the Initialization Display section of the IP Dialog.

Once you have keyed the command, press Enter to start the Initialization Display section of the IP Dialog.

The first screen displayed is the Welcome Screen.

```
*****
***                                     ***
***      Welcome to the VISION:Builder      ***
***                                     ***
***      Installation Preparation Dialog      ***
***                                     ***
***      Press ENTER to Continue            ***
***                                     ***
*****
***
```

Press Enter to see the High-Level Index Information Screen. This will show you the Input Parameter you entered.

```
The High-Level Index Name you provided is:

      > BUILDER.R140 <

This value will be used to reference all the data sets
that you "UNLOADED" from the VISION:Builder System Tape
in the "COPY1" and "COPY2" Jobs.

      Several of the "unloaded" data sets are the
      SMP/E Indirect Libraries referenced in the
      MCS (Modification Control Statements)
      instructions used in the SMP/E process.

Is this High-Level Index Name correct?  Enter Y or N
```

If you did not enter the input parameter with the EXEC command, you will see the same information screen with the default value displayed.

You need to respond to this screen with a Y or N in order to continue the IP Dialog.

If you respond with Y, indicating that the high-level index name is correct, you continue with the Panel Display section of the IP Dialog. The high-level index is used as a prefix to form the Data Set Names (DSNs) of the files unloaded from the system tape. The Panel Display section of the IP Dialog uses several of the unloaded data sets.

There is an intervening display telling you that the Panel Display section of the IP Dialog is about to start. The TSO/ISPF processing load on your system determines how long the start-up takes. If your system is very fast, you might not be able to see the intervening display.

```
Data Sets are being allocated.  
The Panel Display Section will start in a few moments.....
```

Here are some other screens that may or may not display during the Initialization Display section of the IP Dialog, based on your choices, your input, or some processing conditions.

If you respond with N to the High-Level Index Information Screen, you are given the chance to key another value.

```
The High-Level Index Name you provided is:  
  
      > BUILDER.R140 <  
  
This value will be used to reference all the data sets  
that you "UNLOADED" from the VISION:Builder System Tape  
in the "COPY1" and "COPY2" Jobs.  
  
      Several of the "unloaded" data sets are the  
      SMP/E Indirect Libraries referenced in the  
      MCS (Modification Control Statements)  
      instructions used in the SMP/E process.  
  
Is this High-Level Index Name correct?  Enter Y or N  
  
n  
  
ENTER the High-Level Index you would like to use.  
IF Blank, The Default Value used will be > BUILDER.R140 <
```

You key in your new High-Level Index Name or key nothing (leave blank for the default), and press Enter.

The High-Level Index Information Screen displays again for you to verify your input. You have the opportunity again to respond Y or N.

If you enter an invalid value, such as something inappropriate for High-Level Index Names, you will see a display indicating an error condition and asking for a new value.

```
The High-Level Index value you provided is:
      > 123 56 <
it contains invalid characters.
1ST POS   = A-Z or $, #, @ (National Std)
2ND to nTH = A-Z or $, #, @ or 1-9 or . (Period)
ENTER the High-Level Index you would like to use.
IF Blank, The Default Value used will be > BUILDER.R140 <
```

During the transition from the Initialization Display section to the Panel Display section, the high-level index is used as a prefix to form the DSNs of the data sets used by the IP Dialog. If there is a problem with the data sets, error displays describe the condition. Here is a sample of one such display.

```
Data Sets are being allocated.
The Panel Display Session will start in a few moments.....
'BUILDR.R140.PREP.MSGS' DATASET NOT FOUND, Dialog will not run
'BUILDR.R140.PREP.PANELS' DATASET NOT FOUND, Dialog will not run
'BUILDR.R140.PREP.SKELS' DATASET NOT FOUND, Dialog will not run
***-----***
***                THE DIALOG HAS TERMINATED WITH AN ERROR.                ***
***                PRESS THE ENTER KEY                                     ***
***-----***
***
```

Press Enter and the IP Dialog terminates, returning you to the TSO/ISPF Option 6 Screen.

There are several other conditional displays that are self-explanatory. Most likely, you will never see these other displays.

## Panel Display

The Panel Display section of the IP Dialog is where you review and provide values for the variables that are placed into the JCL and control statements that are used during the installation, customization, and maintenance activities described in this book.

The first panel displayed in this section of the IP Dialog is the introduction panel. This panel provides some general information that is supplemented with more details by using the standard HELP facility.

```
BLINT1P  ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ==>

                VISION:Builder Release 14.0

This Dialog will help you prepare all the JCL and Control Statements
needed for the SMP/E Installation, the Customizing, and the SMP/E
Maintenance of your VISION:Builder Software System and its Components.

This Dialog will present all the variables used within the JCLs and
Control Statements that are needed to complete the SMP/E Installation,
the Customizing, and the SMP/E Maintenance Tasks associated with your
VISION:Builder Software System.

Standard Helps and detailed descriptions are associated with each
panel presented by the Dialog. The same information is also provided
in the VISION:Builder Installation Guide.

This Dialog can be rerun at any time. All information entered during
a session is saved in your profile variables. These values will be
redisplayed in subsequent sessions.

Press ENTER to get started.
```

Essentially, you provide information to be merged into the JCL and control statement models that are stored in the BUILDER.R140.PREP.SKELS data set. The resulting File Tailored members are placed in the BUILDER.R140.PREP.JCLCNTL data set, ready for use during the installation, customization, and maintenance tasks described in this book.

**Note:** The BUILDER.R140.PREP.JCLCNTL data set is referenced throughout the installation, customization, and maintenance processes described in this book. The reference is stated as follows:

the PDS data set (...PREP.JCLCNTL).

To start the Panel Display Section of the IP Dialog, press Enter.

## Variables Panels

### Variables Part 1

The Variables Part 1 panel is the first input panel displayed.

```

BLVAR1P  ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ==>

          VARIABLES - Part 1 - SMP/E CSI Information

The OS/390 SMP/E Facility is designed to manage the Installation of
Software Products and track any modifications. SMP/E uses the CSI
(Consolidated Software Inventory) to keep records about the software.

Please provide the SMP/E CSI and SMPPTS Information:  (Use HELP for details)

Will the CSI be New ?      => YES          Enter YES or NO
CSI Data Set Name (...CSI) => BUILDER.R140.CSI
CSI VOLUME   ("NEW" CSI) =>                (If Blank, No VOLUME Used)
SMPPTS - High-Level Index => BUILDER.R140
SMPPTS UNIT  ("NEW" CSI) => SYSDA
SMPPTS VOLSER ("NEW" CSI) =>                (If Blank, No VOLSER Used)


Use ENTER  to Process the Information
Use END    to Save the Information and Exit the Dialog
Use CANCEL to Exit the Dialog Without Saving the entered Information
  
```

This panel contains information needed for defining the SMP/E CSI (Consolidated Software Inventory) data sets for VISION:Builder. These CSI data sets are used by SMP/E to contain all the information needed to manage and track your installed software.

The following is a description of the entries on the Variables Part 1 Panel.

Will the CSI be New ?	Required. Enter YES or NO to indicate whether new CSI Data Sets should be defined to hold VISION:Builder information. In the basic SMP/E approach described in this book, you will setup a new CSI to keep track of VISION:Builder.
CSI Data Set Name (...CSI)	Required. Enter the data set name of your CSI VSAM Cluster. The default is the High-Level Index you supplied during the Initialization Display Section of the IP Dialog suffixed with .CSI. When a new CSI is being created, this will be the VSAM Cluster Name.
	If you will be using an existing CSI, supply the Cluster name here.

CSI VOLUME ("NEW" CSI)	Optional. This entry only applies to a new CSI. This entry will be used in the VOLUME parameter of the VSAM Cluster definition for the new CSI. If left blank, no VOLUME parameter will be specified.
SMPPTS - High-Level Index	Required. Enter the High-Level Index of your SMPPTS data set name. The PTS is used by SMP/E to temporarily store SYSMODS waiting to be installed. The default is the High-Level Index you supplied during the Initialization Display Section of the IP Dialog. When a new CSI is being created, this High-Level Index will be suffixed with .SMPPTS to form the data set name of the newly created PTS data set.  If you will be using an existing CSI and PTS, supplied the appropriate High-Level Index name here.
SMPPTS UNIT ("NEW" CSI)	Optional. This entry only applies when a new CSI and PTS are being created. This is the UNIT for the new PTS data set.
SMPPTS VOLSER ("NEW" CSI)	Optional. This entry only applies when a new CSI and PTS are being created. This will be used in the VOL=SER= parameter. If left blank, no VOL=SER= will be specified in the JCL.

After you have keyed in the necessary information, press Enter to edit and save the information.

A Variables Part 1 panel displays again, showing you the information you supplied and indicating that it was saved.

```

BLVAR1PA ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ===>
The Information was saved, Use ENTER to continue, Use END to Re-Input.
      VARIABLES - Part 1 - SMP/E CSI Information

The OS/390 SMP/E Facility is designed to manage the Installation of
Software Products and track any modifications. SMP/E uses the CSI
(Consolidated Software Inventory) to keep records about the software.

Please provide the SMP/E CSI and SMPPTS Information:  (Use HELP for details)

Will the CSI be New ?      => YES          Enter YES or NO
CSI Data Set Name (...CSI) => BUILDER.R140.CSI
CSI VOLUME      ("NEW" CSI) =>              (If Blank, No VOLUME Used)
SMPPTS - High-Level Index => BUILDER.R140
SMPPTS UNIT     ("NEW" CSI) => SYSDA
SMPPTS VOLSER   ("NEW" CSI) =>              (If Blank, No VOLSER Used)

Use ENTER to Continue the Dialog with the Next Display
Use END   to Return to Input Mode for this Display
Use CANCEL to Exit the Dialog, the entered Information has been Saved

```

- If the information is correct, press Enter to continue the IP Dialog with the next panel.
- To change the information, use End (or PF3/F3) to go back to the Variables Part 1 Input Panel.

### Variables Part 2

Once you have completed the Variables Part 1 panel, you go to the Variables Part 2 panel.

```
BLVAR2P ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ==>

          VARIABLES - Part 2 - SMP/E Work Data Sets Information

The OS/390 SMP/E Facility uses several Work Data Sets during the
various processes. These work Data Sets are:
  SMPMTS, SMPSCDS, SMPSTS, SMPLOG, SMPLOGA.

Please provide the Work Data Sets Information:  (Use HELP for details)

High-Level Qualifier      => BUILDER.R140
Work Disk UNIT            => SYSDA
Work Disk VOLSER          =>                (If Blank, No VOLSER Used)
SMPTLIB Disk UNIT         => SYSDA
SMPTLIB Disk VOLSER       =>                (If Blank, No VOLSER Used)


Use ENTER  to Process the Information
Use END    to Save the Information and Goto the Previous Display
Use CANCEL to Exit the Dialog Without Saving the entered Information
```

This panel contains information needed for the various SMP/E work data sets. These work data sets are used for staging or backing up elements during SMP/E processing and activities.

The following is a description of the entries on the Variables Part 2 Panel.

High-Level Qualifier	Required. Enter the High-Level Qualifier for the work data sets. The default is the High-Level Index you supplied during the Initialization Display Section of the IP Dialog. When the work data sets are created, this High-Level Index will be suffixed with SMP/E Work Data Set DD Name.
Work Disk UNIT	Required. Enter the UNIT value to be used when Work Data Sets are created.
Work Disk VOLSER	Optional. Enter the VOLSER value to be used when Work Data Sets are created. This will be used in the VOL=SER= parameter. If left blank, no VOL=SER= will be specified.
SMPTLIB Disk UNIT	Required. Enter the UNIT value to be used when SMPTLIB Work Data Sets are created.

SMPTLIB Disk  
VOLSER      Optional. Enter the VOLSER value to be used when SMPTLIB Work Data Sets are created. This will be used in the VOL=SER= parameter. If left blank, no VOL=SER= will be specified.

After you have keyed in the necessary information, press Enter to edit and save the information.

A Variables Part 2 Panel displays again, showing you the information you supplied, and indicating that it was saved.

```

BLVAR2PA ----- VISION:Builder Installation Preparation Dialog -----
OPTION ==>
The Information was saved, Use ENTER to continue, Use END to Re-Input.
      VARIABLES - Part 2 - SMP/E Work Data Sets Information

The OS/390 SMP/E Facility uses several Work Data Sets during the
various processes. These work Data Sets are:
  SMPMTS, SMPSCDS, SMPSTS, SMPLOG, SMPLOGA.

Please provide the Work Data Sets Information:  (Use HELP for details)

High-Level Qualifier      => BUILDER.R140
Work Disk UNIT            => SYSDA
Work Disk VOLSER          =>                (If Blank, No VOLSER Used)
SMPTLIB Disk UNIT         => SYSDA
SMPTLIB Disk VOLSER       =>                (If Blank, No VOLSER Used)


Use ENTER  to Continue the Dialog with the Next Display
Use END    to Return to Input Mode for this Display
Use CANCEL to Exit the Dialog, the entered Information has been Saved

```

- If the information is correct, press Enter to continue the IP Dialog with the next panel.
- To change the information, use END (or PF3/F3) to go back to the Variables Part 2 Input Panel.

**Variables Part 3**

The next display is the Variables Part 3 panel.

```
BLVAR3P  ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ==>

          VARIABLES - Part 3 - SMP/E Product Information

The OS/390 SMP/E Facility uses Zones to maintain your installed Product
and its associated information. The Global Zone contains processing
related information and indexes to the Distribution and Target Zones.
The Distribution and Target Zones contain processing related information
about the elements in the Distribution and Target Libraries.

Please provide the Product Information:  (Use HELP for details)

Distribution Zone Name      => BL140DZ
Target Zone Name           => BL140TZ
High-Level Qual for Libs   => BUILDER.R140
Distribution Libs UNIT      => SYSDA
Distribution Libs VOLSER    =>                (If Blank, No VOLSER Used)
Target Libs UNIT           => SYSDA
Target Libs VOLSER         =>                (If Blank, No VOLSER Used)

Use ENTER  to Process the Information
Use END    to Save the Information and Goto the Previous Display
Use CANCEL to Exit the Dialog Without Saving the entered Information
```

This panel requests information that SMP/E uses to establish a structure for the management and control of the VISION:Builder elements.

The following is a description of the entries on the Variables Part 3 Panel.

Distribution Zone Name	Required. The SMP/E Name assigned to identify the Distribution Zone.
Target Zone Name	Required. The SMP/E Name assigned to identify the Target Zone.

High-Level Qual  
for Libs

Required. Enter the High-Level Qualifier for the Distribution and Target Libraries. The default is the High-Level Index you supplied during the Initialization Display Section of the IP Dialog. When the Library data sets are created, this High-Level Index will be suffixed with a VISION:Builder standard name.

Using the default High-Level Index, here are the library names that will be created:

Distribution	BUILDER.R140.SMPE.D.BLSYSL
Libraries	BUILDER.R140.SMPE.D.BLSAMP
	BUILDER.R140.SMPE.D.WBCLIST
	BUILDER.R140.SMPE.D.WBMSGs
	BUILDER.R140.SMPE.D.WBPANEL
	BUILDER.R140.SMPE.D.WBSKELS
Target Libraries	BUILDER.R140.SMPE.T.BLSYSL
	BUILDER.R140.SMPE.T.BLSAMP
	BUILDER.R140.SMPE.T.WBCLIST
	BUILDER.R140.SMPE.T.WBMSGs
	BUILDER.R140.SMPE.T.WBPANEL
	BUILDER.R140.SMPE.T.WBSKELS

Here is a brief description of the library contents:

.BLSYSL	Load Modules for VISION:Builder and its components. Includes the VISION:Builder Engine, the COMLIB Component, and the Workbench for ISPF Component.
.BLSAMP	Source Code, Control Statements, and so on, used to customize your software system.
.WBCLIST	Workbench for ISPF CLIST members.
.WBMSGs	Workbench for ISPF Messages members.
.WBPANEL	Workbench for ISPF Panel and Help members.
.WBSKELS	Workbench for ISPF Skeleton members.

Distribution  
Libs UNIT

Required. Enter the UNIT value to be used when the Distribution Libraries are created.

Distribution  
Libs VOLSER

Optional. Enter the VOLSER value to be used when the Distribution Libraries are created. This will be used in the VOL=SER= parameter. If left blank, no VOL=SER= will be specified.

Target Libs  
UNIT

Required. Enter the UNIT value to be used when the Target Libraries are created.

Target Libs  
VOLSER

Optional. Enter the VOLSER value to be used when the Target Libraries are created. This will be used in the VOL=SER= parameter. If left blank, no VOL=SER= will be specified.

After you have keyed in the necessary information, press Enter to edit and save the information.

A Variables Part 3 Panel displays again showing you the information you supplied, and indicating that it was saved.

```
BLVAR3PA ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ==>
The Information was saved, Use ENTER to continue, Use END to Re-Input.
      VARIABLES - Part 3 - SMP/E Product Information

The OS/390 SMP/E Facility uses Zones to maintain your installed Product
and its associated information. The Global Zone contains processing
related information and indexes to the Distribution and Target Zones.
The Distribution and Target Zones contain processing related information
about the elements in the Distribution and Target Libraries.

Please provide the Product Information:  (Use HELP for details)

Distribution Zone Name      => BL140DZ
Target Zone Name           => BL140TZ
High-Level Qual for Libs   => BUILDER.R140
Distribution Libs UNIT      => SYSDA
Distribution Libs VOLSER    =>                (If Blank, No VOLSER Used)
Target Libs UNIT           => SYSDA
Target Libs VOLSER         =>                (If Blank, No VOLSER Used)

Use ENTER  to Continue the Dialog with the Next Display
Use END    to Return to Input Mode for this Display
Use CANCEL to Exit the Dialog, the entered Information has been Saved
```

- If the information is correct, press Enter to continue the IP Dialog with the next panel.
- To change the information, use END (or PF3/F3) to go back to the Variables Part 3 Input Panel.

## Variables Part 4

The next display is the Variables Part 4 panel.

```

BLVAR4P  ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ==>

          VARIABLES - Part 4 - Miscellaneous OS/390 Information

Additional information is needed to accommodate the various tasks that are
performed during the Installation, Customizing and Maintenance activities.

Please provide the following Information:      (Leave BLANK if none exists)

IBM System Maclib (Req'd)  => SYS1.MACLIB
IBM LE  Run  Library       => CEE.SCEERUN
IBM DB2 Load Library      => DB2.SDSNLOAD
IBM IMS Res  Library       => IMS.RESLIB


Use ENTER  to Process the Information
Use END    to Save the Information and Goto the Previous Display
Use CANCEL to Exit the Dialog Without Saving the entered Information

```

This panel contains OS/390 information that helps prepare the JCL and control statements.

The following is a description of the entries on the Variables Part 4 Panel.

IBM System Maclib (Req'd)	Required. The name of the IBM System Macros Library.
IBM LE Run Library	Optional. The name of the IBM LE (Language Environment) Runtime (Linklib) Library. If no such library exists, leave the entry blank.
IBM DB2 Load Library	Optional. The name of the IBM DB2 Load Library. If no such library exists, leave the entry blank.
IBM Res Library	Optional. The name of the IBM IMS Resident Library. If no such library exists, leave the entry blank.

After you have keyed in the necessary information, press Enter to edit and save the information.

A Variables Part 4 Panel displays again, showing you the information you supplied, and indicating that it was saved.

```
BLVAR4PA ----- VISION:Builder Installation Preparation Dialog -----
OPTION  ==>
The Information was saved, Use ENTER to continue, Use END to Re-Input.
      VARIABLES - Part 4 - Miscellaneous OS/390 Information

Additional information is needed to accommodate the various tasks that are
performed during the Installation, Customizing and Maintenance activities.

Please provide the following Information:      (Leave BLANK if none exists)

IBM System MacLib (Req'd)  => SYS1.MACLIB
IBM LE  Run  Library      => CEE.SCEERUN
IBM DB2 Load Library      => DB2.SDSNLOAD
IBM IMS Res  Library      => IMS.RESLIB


Use ENTER  to Continue the Dialog with the Next Display
Use END    to Return to Input Mode for this Display
Use CANCEL to Exit the Dialog, the entered Information has been Saved
```

If the information is correct, press Enter to continue the IP Dialog with the next panel.

To change the information, use END (or PF3/F3) to go back to the Variables Part 4 panel.

We have completed the Variables portion in the Panel Display Section of the IP Dialog.

## JCL Panels

The next portion of the IP Dialog requests the JOB card information and indicates where the tailored JCL and Control Statements will be placed.

### JCL Part 1

The next display is the JCL Part 1 panel.

```
BLJCL1P ----- VISION:Builder Installation Preparation Dialog -----
COMMAND ===>

          Generate JCL - Part 1 - Setup the JOB Control Statements

The JCL and Jobs that will be created for the Installation, Customizing
and Maintenance Tasks associated with your VISION:Builder System and
its Components are ready to be generated.

Please provide the JOB Control Statement information for the JCL:

Your JOB Name      =>          Leave BLANK and default Job Names will be used
JOB Stmt Info      => (ACCT)
Additional Stmts:
//*
//*
//*
//*
//*

Use ENTER to Process the Information
Use END   to Save the Information and Goto the Previous Display
Use CANCEL to Exit the Dialog Without Saving the entered Information
```

This panel contains the job statement information. The job statements start each JCL set that is created by the tailoring process.

The following is a description of the entries on the JCL Part 1 panel.

Your JOB Name	Optional. This name will be used to form a JOB Name for each JCL that is created by this Dialog Process. The name will be suffixed with a Character (a number or letter) to form the complete JOB Name. You can leave the JOB Name blank, and the system will supply a default JOB Name that will match the member name of the created JCL.
JOB Stmt Info	Optional. This is the JOB Statement information, generally your accounting information.
Additional Stmts:	Optional. There is room for five addition JCL statements in the JOB Statement group. These can be whatever you need. Remember, these will follow the JOB Statement and precede all other JCL statements. If you blank out a line, it will not be part of the statements in the JOB group.

After you have keyed in the necessary information, press Enter to edit and save the information.

### JCL Part 2

The JCL Part 2 panel displays next showing you what your JOB Statement group will look like.

```
BLJCL2P ----- VISION:Builder Installation Preparation Dialog -----
COMMAND ==>
The Information was saved, Use ENTER to continue, Use END to Re-Input.
Generate JCL - Part 2 - Verify the JOB Control Statements

The JCL and Jobs that will be created for the Installation, Customizing
and Maintenance Tasks associated with your VISION:Builder System and
its Components are ready to be generated.

Here are the JOB Control Statements that will start each JCL Member.

/*DEFAULT* JOB (ACCT)
/**
/**
/**
/**
/**

Use ENTER to Continue the Dialog with the Next Display
Use END to Return to Input Mode for this Display
Use CANCEL to Exit the Dialog, the entered Information has been Saved
```

- If the information is correct, press Enter to continue the IP Dialog with the next panel.
- To change the information, use END (or PF3/F3) to go back to the JCL Part 1 Input Panel.

### JCL Part 3

The JCL Part 3 panel displays next.

```
BLJCL3P ----- VISION:Builder Installation Preparation Dialog -----
COMMAND ==>

Generate JCL - Part 3 - Create the JCL Job Members

The JCL and Control Statements that will be used during the Installation,
Customizing and Maintenance of your VISION:Builder System will now be
created using the variables established during the preceding dialog.
The members will be placed into a PDS for subsequent use as described
in this book.

The PDS Data Set Name for the members is:

Data Set Name => BUILDER.R140.PREP.JCLCNTL

This Data Set was created and cataloged when the VISION:Builder
System Tape Files were copied to Disk by JOB BLCOPY2. It contains
Members that are used during the JCL Job Create Process that follows.

(See the VISION:Builder Installation Guide, Step 2 for details.)

Use ENTER to Continue
Use END to Save the Information and Goto the Previous Display
Use CANCEL to Exit the Dialog Without Saving the entered Information
```

This panel shows you the name of the PDS data set (...PREP.JCLCNTL) where the tailored JCL and Control Statement members will be placed. This data set should already exist since it was created during the System Tape Unload. See Chapter 2 Step 2 for details.

This panel is here as a reminder and as an indication that the data set can be accessed without any problems. The Dialog checks the data set format characteristics for consistency.

**Note:** The BUILDER.R140.PREP.JCLCNTL data set will be referenced throughout the Installation, Customizing and Maintenance processes described in this book. The reference will be stated as “the PDS Data Set (...PREP.JCLCNTL)”.

The next display after the JCL Part 3 panel is usually the [JCL Part 5](#) panel.

#### JCL Part 4

The JCL Part 4 panel only displays if there is an error while accessing the PDS data set (...PREP.JCLCNTL). If an error occurs, you would see the JCL Part 4 panel, indicating the error condition and telling you the IP Dialog will be terminated. You need to correct the situation and restart the IP Dialog. Remember that all the information you have entered up to this point has been saved and will be redisplayed during your next session. Just press Enter to quickly jump through the displays.

The following is an example of the JCL Part 4 Panel showing an error.

```
BLJCL4P  ----- VISION:Builder Installation Preparation Dialog -----
COMMAND ===>

          Generate JCL - Part 4 - ERROR Accessing Install JCL Data Set

There was an ERROR Accessing the '...PREP.JCLCNTL' Data Set.

This Data Set should have been created and cataloged when the VISION:Builder
System Tape Files were copied to Disk by JOB BLCOPY2. The Data Set contains
Members that are Needed for the JCL Job Create Process and Installation.
(See the VISION:Builder Installation Guide, Step 2 for details.)

Please Correct the problem and Restart the Installation Preparation Dialog.

The Data Set Name  => MARKIV.JK14.PREP.JCLCNTL
                   The PDS for the Tailored Install JCL is NOT Available.
                   The reason is DATASET NOT FOUND . The Dialog will be
                   Terminated.

Use ENTER  to Exit the Dialog, all entered information is Saved
Use END    to Exit the Dialog, all entered information is Saved
Use CANCEL to Exit the Dialog, all entered information is Saved
```

A termination screen displays after the JCL Part 4 panel.

```
*****
***                                     ***
***   The Dialog was "TERMINATED" due to an ERROR   ***
***                                     ***
*****

***
```

At this point, you would press Enter to return to the TSO/ISPF Option 6 Display.

### JCL Part 5

The last screen in the IP Dialog is the JCL Part 5 panel.

```
BLJCL5P  ----- VISION:Builder Installation Preparation Dialog -----
COMMAND ==>

        Generate the JCL and Control Statements

The JCL and Control Statements will now be generated

        by using the

        ISPF File Tailoring Facility.

Press ENTER to Start the File Tailoring Process.

* * * * *
* The process will take a few moments to complete *
* * * * *

Use ENTER  to Start the File Tailoring Process
Use END    to Go to the Previous Display
Use CANCEL to Exit the Dialog, the entered Information has been Saved
```

This panel is just an information display that tells you the ISPF file tailoring process will start as soon as you press Enter. There is a note reminding you that this will take a few moments to complete, so be patient.

When the file tailoring is completed, all the prepared JCL and Control Statement Members have been added or replaced into the PDS data set (...PREP.JCLCNTL), ready for you to use in the actual installation, customization and maintenance processes described in [Chapter 4, SMP/E Setup and Basic Installation](#), [Chapter 5, Customizations and Setups](#), and [Chapter 6, Maintenance and Support](#).

The final screen tells you that the you have completed the process and that the prepared members are in the named data set.

```
*****
***                                     ***
***      THANK YOU. You have COMPLETED the      ***
***                                     ***
*** VISION:Builder Installation Preparation Dialog. ***
***                                     ***
*** Now refer to the Installation Guide for the ***
*** instructions on Running the prepared JOBS. ***
***                                     ***
*** The prepared items are in the data set named ***
***                                     ***
*** > BUILDER.R140.PREP.JCLCNTL < ***
***                                     ***
*****
***
```

Now you are ready to perform the SMP/E setup and the basic VISION:Builder installation, described in [Chapter 4, SMP/E Setup and Basic Installation](#).



# SMP/E Setup and Basic Installation

The system tape files unloaded to disk data sets in Steps 1 and 2 are described in Chapter 2. The JCL and control statements you need for the SMP/E setup and basic VISION:Builder installation were prepared in Step 3 using the ISPF Installation Preparation Dialog described in Chapter 3.

At this point, you need to run nine job streams to establish and define the SMP/E CSI and zones, and install the basic VISION:Builder. These job streams are in the PDS data set (...PREP.JCLCNTL) and the default name is BUILDER.R140.PREP.JCLCNTL.

The following members contain these nine job streams for the basic VISION:Builder installation.

Member Name	Function of the Job within This Member
BLSMPE#1	Allocate the SMP/E CSI, the SMP/E work data sets and all the associated data sets for the distribution and target libraries.
BLSMPE#2	Define the SMP/E global, distribution and target zones.
BLSMPE#3	Receive the MCS control statements and SYSMODs into the SMP/E global zone and work data sets.
BLSMPE#4	Receive the PTF and APAR SYSMODs into the SMP/E global zone and work data sets.
BLSMPE#5	Apply the SYSMODS (modules and elements) to the target libraries.
BLSMPE#6	Apply the SYSMODS (PTFs) to the target libraries.
BLSMPE#7	Run the Installation Verification Procedure using the target load library.
BLSMPE#8	Accept the SYSMODS (modules and elements) to the distribution libraries.
BLSMPE#9	Apply the SYSMODS (PTFs) to the distribution libraries.

The following members are additional members in the PDS data set (...PREP.JCLCNTL) that are referenced within the above jobs. These are control statements and SYSMODS (PTFs and APARs).

Member Name	Description
BLSMCS#0	MCS Statements - the VISION:Builder software system function.
BLSMCS#1	MCS Statements - the VISION:Builder engine elements.
BLSMCS#2	MCS Statements - the VISION:Builder COMLIB component elements.
BLSMCS#3	MCS Statements - the VISION:Builder Workbench™ for ISPF component elements.
BLSMCS#4	MCS Statements - the SAS/C Link Lib (runtime) elements.
CCVC140 CCVPE00	The JCLIN (IEBCOPYs) for the distribution and target load libraries.
PTFS	The latest SYSMODS (PTFs) for VISION:Builder.
APARS	The latest SYSMODS (APARs) for VISION:Builder.

## Step 4 – Allocate Data Set

In Step 4, you allocate all the data sets needed by SMP/E to manage, control and maintain VISION:Builder and its components. This includes the SMP/E CSI, the associated work data sets, and the distribution and target libraries.

Using the JCL in member BLSMPE#1 in the PDS data set (...PREP.JCLCNTL), run the job to allocate the data sets.

Any existing or previously defined data sets of the same names are deleted before the new data sets are allocated.

The following data sets are allocated. They are shown with the default high-level index. Check the JCL in member BLSMPE#1 for your high-level index values, if they differ from the default.

BUILDER.R140.CSI	BUILDER.R140.SMPE.T.BLSYSL
BUILDER.R140.CSI.DATA	BUILDER.R140.SMPE.T.WBCLIST
BUILDER.R140.CSI.INDEX	BUILDER.R140.SMPE.T.WBMSG
BUILDER.R140.SMPE.D.BLSAMP	BUILDER.R140.SMPE.T.WBPANEL
BUILDER.R140.SMPE.D.BLSYSL	BUILDER.R140.SMPE.T.WBSKELS
BUILDER.R140.SMPE.D.WBCLIST	BUILDER.R140.SMPLOG
BUILDER.R140.SMPE.D.WBMSG	BUILDER.R140.SMPLOGA
BUILDER.R140.SMPE.D.WBPANEL	BUILDER.R140.SMPMTS
BUILDER.R140.SMPE.D.WBSKELS	BUILDER.R140.SMPPTS
BUILDER.R140.SMPE.T.BLSAMP	BUILDER.R140.SMPSCDS

## Step 5 – Define the CSI and the Global, Distribution, and Target Zones

In Step 5, you define the VISION:Builder global, distribution, and target zones into the CSI. This is the information needed by SMP/E to manage, control, and maintain VISION:Builder.

Using the JCL in member BLSMPE#2 in the PDS data set (...PREP.JCLCNTL), run the job to define the VISION:Builder global, distribution, and target zones into the CSI.

## Step 6 – RECEIVE the MCS and SYSMODS into the Global Zone

In Step 6, you RECEIVE the Modification Control Statements (MCS) and VISION:Builder software system elements (SYSMODS) into the global zone and SMP/E data sets.

Using the JCL in member BLSMPE#3 in the PDS data set (...PREP.JCLCNTL), run the job to RECEIVE VISION:Builder into the global zone and SMP/E data sets.

## Step 7 – RECEIVE the PTF and APAR SYSMODS into the Global Zone

In Step 7, you RECEIVE the PTF and APAR SYSMODS into the global zone and SMP/E data sets.

**Note:** Prior to VISION:Builder Release 14.0, PTFs were known as SMs or GSMs (General System Modifications). These types of patches apply to all systems and correct or enhance the software system.

**Note:** Prior to VISION:Builder Release 14.0, APARs were known as RSMs (Restricted System Modifications). These type of patches only apply, if at all, to sites with unique requirements. See Chapter 5, Step 13, for more information regarding APARs.

Using the JCL in member BLSMPE#4 in the PDS data set (...PREP.JCLCNTL), run the job to RECEIVE the VISION:Builder software system PTF and APAR SYSMODS into the global zone and SMP/E data sets.

## Step 8 – APPLY the VISION:Builder Elements (SYSMODS) to the Target Libraries

In Step 8, you APPLY the VISION:Builder software system elements (SYSMODS) into the target libraries.

Using the JCL in member BLSMPE#5 in the PDS data set (...PREP.JCLCNTL), run the job to APPLY VISION:Builder into the target libraries. This creates the VISION:Builder system load library in the target libraries.

## Step 9 – APPLY the VISION:Builder PTF SYSMODS to the Target Libraries

In Step 9, you APPLY the VISION:Builder software system PTF SYSMODS into the target libraries.

Using the JCL in member BLSMPE#6 in the PDS data set (...PREP.JCLCNTL), run the job to APPLY the VISION:Builder PTFs into the target libraries.

At this point, the Target Libraries contain the default VISION:Builder software system.

## Step 10 – Run the Installation Verification Procedure using the Target Load Library.

In Step 10, you Run a Job that verifies that the basic VISION:Builder elements are installed and operational.

Using the JCL in member BLSMPE#7 in the PDS data set (...PREP.JCLCNTL), run the job to verify the Install. This simple job exercises several different functions of VISION:Builder using the target load library.

There will be “No” permanent data sets created by this job stream. The VISION:Builder uses the IBM Language Environment (LE, formerly LE/370). The LE runtime library must be available when running VISION:Builder jobs either through the Linklist or a concatenation to the JOBLIB (or STEPLIB) DD statements.

The job stream contains the following steps. Each step completes with a condition code zero (0000).

Step	Description
INIT	A library utility run to allocate and initialize the M4LIB common library.
DEFRUN1	A definition run that catalogs a file and a table definition in the M4LIB.
PROCRUN	A single step processing run to read an in-stream data file and produce two reports. The cataloged file and table definitions are utilized from the M4LIB. The Advanced Syntax Language (ASL) translator is invoked. Your standard SORT program is called to sort the report data. The input file data is processed and two reports are produced.
LIBRUN	A library utility run to back up (dump), reinitialize, and restore the contents of the M4LIB. This is a process that condenses the M4LIB.
DEFRUN2	A definition run to produce glossary listings of the file and table definitions.

After completing this step, your basic installation into the target libraries is complete and verified and you can now ACCEPT the elements into the distribution libraries.

## Step 11 – ACCEPT the VISION:Builder Elements (SYSMODS) to the Distribution Libraries

In Step 11, you ACCEPT the VISION:Builder software system elements (SYSMODS) into the distribution libraries.

Using the JCL in member BLSMPE#8 in the PDS data set (...PREP.JCLCNTL), run the job to ACCEPT VISION:Builder into the distribution libraries. This creates the VISION:Builder system load library in the distribution libraries.

## Step 12 – ACCEPT the VISION:Builder PTF SYSMODS to the Distribution Libraries

In Step 12, you ACCEPT the VISION:Builder PTF SYSMODS into the distribution libraries.

Using the JCL in member BLSMPE#9 in the PDS data set (...PREP.JCLCNTL), run the job to ACCEPT the VISION:Builder PTFs into the distribution libraries.

At this point, both the distribution libraries and the target libraries contain the default VISION:Builder software system.

See [Chapter 5, Customizations and Setups](#) for information about reviewing and determining the customizations and setups that apply to your use of VISION:Builder.

# Customizations and Setups

---

In a software system as diverse as VISION:Builder, there are several capabilities and facilities that can be customized, setup, and tailored to meet specific environmental and operational requirements. The steps described in this chapter are optional, and the ones that apply to your site are dependent on how the software system and components are used.

- All the customizations are done outside of SMP/E control, except the APAR processing.
- All the customizations are directed to the target libraries.
- All the customizations affect independent modules, except the APAR processing.

Keep track of the various customization jobs that you perform. This may be important if you ever need to go back to the default VISION:Builder software system and rebuild your customized system.

## Step 13 – APPLY Customizing APARs

In VISION:Builder, an SMP/E APAR is a customization to the system that satisfies a unique site requirement. Prior to Release 14.0, these patches were known as RSMs (Restricted System Modifications). If you previously installed RSMs with VISION:Builder, you may need to install the current release of the APARs into your new VISION:Builder Release 14.0 system.

You can easily identify if any APAR/RSM modifications have been previously applied to the VISION:Builder engine and COMLIB component by looking at the top portion of the VISION:Builder banner page that precedes a source listing at the M4LIST DD. Any modification number that is less than 200 is an APAR/RSM and a candidate for a comparable upgraded APAR for the current release. (The modification numbers starting at 200 are the PTFs/GSMs.)

In the PDS data set (...PREP.JCLCNTL), the member BLXBAN#1 shows a simple batch job that produces a banner page from the newly installed system. By changing the JOBLIB to point to a previous version of VISION:Builder, you can get a banner page from that version of the system for comparison purposes.

Here is an example of a Release 14.0 banner page:

```
JAN 31, 2002 19.49.51                                     PAGE      1

*****
* VISION:Builder  4400  (OS/390 - 14.0) *
*      COPYRIGHT (C) 2002
* COMPUTER ASSOCIATES INTERNATIONAL, INC. *
*****

BUILDER DEFAULT VERSION      ENABLED FOR IBM LANGUAGE ENVIRONMENT      BUILD STAMP = 102031,17:51:05.

BUILDER MODIFICATIONS (PTFs, APARs) INSTALLED
101,125,131,151,201,202,203
COMLIB MODIFICATIONS (PTFs, APARs) INSTALLED
201,202,203,204,205,206,207,208,209,210,212,213,214,215

=====
=
=          I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )
=
=  SYSTEM DELIMITER:  #          PAGE HEIGHT:      66          M4LIST WIDTH:      132          DEF WIDTH OF PAGE:  0
=
=  AUTO GRAND:        N          HEADING CHAR:      -          SUBTITLE REPEAT:  N          INVALID FIELD:      *
=
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:    (
=
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:    Y
=
=  CONSOLE MESSAGES:   N          MAREPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1
=
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288  DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z
=
=  CURRENCY CHAR:      $          PLUS CHAR:        +          MINUS CHAR:        -          CHECK PROTECT CHAR: *
=
=  DECIMAL CHAR:      .          GROUPING CHAR:     ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*
=
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:  _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |
=
=  VERTICAL HASH:      -          MINUTES/HOUR:    60          SECONDS/MINUTE:    60          TIME DELIMITER:    HH:MM:SS
=
=  DATE FORMAT:        MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY  ISDATE DELIMITER:    YYYY-MM-DD  JULIAN DELIMITER:    YY.DDD
=
=  SORT PROGRAM CODE:  2          MINCORE VALUE:    12 K          ALT M4LIST WIDTH:    132          ALT DEF W/OF PAGE:  0
=
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:    0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMNN:  1,024 K
=
=  CONDITION CODE 1:   0          CONDITION CODE 2:  4          CONDITION CODE 3:    8          CONDITION CODE 4:   16
=
=====
```

Figure 5-1 VISION:Builder Release 14.0 Banner Page

In the shaded area you see the BUILDER MODIFICATIONS and COMLIB MODIFICATIONS title lines followed by the PTF and APAR number identifiers. If these title lines are not shown, it means there are no numbers to display, that is, there are no PTFs or APARs installed.

The PTFs and APARs are identified by component and number using the following format:

CCNNNNNN

where:

- CC is the Component Identifier:
  - BL VISION:Builder engine
  - CL COMLIB component
  - WB Workbench for ISPF component

- NNNNN is the Modification Number Identifier:
  - 00001 to 00199 Numbers assigned to APARs, special patches
  - 00200 to 00500 Numbers assigned to PTFs, general patches

The latest versions of the APARs for VISION:Builder, the COMLIB component, and the VISION:Workbench for ISPF component are delivered on the system tape. In [Step 7 – RECEIVE the PTF and APAR SYSMODS into the Global Zone on page 4-4](#), the job to RECEIVE PTFs and APARs into the SMP/E global zone and work data sets made the elements available for APPLY runs as needed.

In the PDS data set (...PREP.JCLCNTL), the members PTFS and APARS contain the current set of control statements for the PTFs and APARs. The control statements contain comments for each item that describe the situation addressed by the PTF or APAR. Review the description of any APAR you are considering for your system. Contact Computer Associates Technical Support if you have any questions, concerns, or if you just need more information regarding an APAR. See [Contacting Computer Associates on page 1-11](#).

You can use the following JCL members, provided in the PDS data set (...PREP.JCLCNTL), to APPLY, RESTORE and ACCEPT the APARs. At most sites, there are ISPF-driven facilities that can just as easily be used to perform these SMP/E processes.

JCL Member	Description
BLSMPE#A	This APPLYs APARs to the target libraries.
BLSMPE#B	This RESTOREs (removes) APARs from the target libraries.
BLSMPE#C	This ACCEPTs APARs to the distribution libraries.

**Note:** Once you ACCEPT an element, such as an APAR or PTF, into the distribution libraries, there is no direct method for restoring the previous version of an element.

The decision to APPLY and ACCEPT APARs is made at your site. As a rule, any APAR (formally RSM) that is applied to your previous release of VISION:Builder is probably a standard part of your system. These would be the APARs that you can comfortably ACCEPT. If you are evaluating a new APAR, you probably want to wait awhile before you ACCEPT the modification into your distribution libraries.

**Note:** The APAR runs get a return code of 4 from APPLY and ACCEPT runs because they do not contain prerequisites for other PTFs and APARs.

When PTF runs are performed after an APAR has been processed, they get a return code of 4 because the PTF will not contain prerequisites for any APARs. Remember, the APARs can be identified by their number, which is in the range of cc00001 to cc00199, with the cc being a component identifier.

## Customization Activity Considerations

After you have completed Step 13, the SMP/E activities are finished. Only the installation of any future PTFs or APARs are done under the control of SMP/E.

All the subsequent customizing, tailoring and setup tasks described in the following sections are done outside the control of SMP/E. These tasks are identical to the procedures used in previous releases of VISION:Builder. The only difference is the need to manage your SMP/E controlled libraries, specifically the Target Libraries.

There are two basic approaches you can take regarding the management of the Target Libraries. The approach you choose depends on how you want to manage the customized elements and your Target Libraries. As a basic rule, you should save the original versions of any elements you change, and you should have a step-by-step procedure describing how and what you customize (so that you can repeat the process, if needed).

The two basic approaches to the management of your customizing activities are:

1. This approach is straight-forward and requires the minimum amount of management intervention. All customizing and tailoring is done directly into the Target Libraries. Any future PTFs also go into the same Target Libraries via the SMP/E APPLY process. The PTFs also go into the Distribution Libraries via the SMP/E ACCEPT process. The Target Libraries (or copies) are then used in production. This is the default approach, and all the JCL and associated control statements in the PDS data set (...PREP.JCLCNTL) have been built for this method. Of course, the changed elements (source, panels, clist, and so on) require saving and you should document the various changes in case they need to be repeated.
2. This approach requires more attention to details and procedures regarding the management of your customizing activities. The premise here is that the Target and Distribution Libraries are synchronized at the point where Step 13 of the installation process is completed, and you will always keep them synchronized.

First, copy the Target Libraries. These copies are where the customizing and tailoring is applied. As you do the customizing, tailoring and setups, you keep step-by-step details of what was done so it can be repeated. Whenever a PTF is put into the Target and Distribution Libraries via the SMP/E APPLY and ACCEPT runs, you rebuild your copied Libraries and repeat the customization. In most cases, only the Load Library is changed via PTFs or APARs. The load modules that you can customize are usually not PTF'd. (The one exception is the MARKIV load module, which you can customize with Static Own Code Integration.) With this in mind, it is possible to keep a copy of the customized load modules (like M4PARAMS) in an auxiliary Load Library, which is then copied (with REPLACE) back into the Main Load Library, after each refresh from the Target Load Library.

Whichever approach you choose, keep track of your activities for reference. See the next section for information and sample Jobs for copying the VISION:Builder System Load Library.

## Copying the VISION:Builder System Load Library

If you need to copy the entire VISION:Builder Load System Load Library, keep in mind that the load library contains a module named MARKIV (and its aliases) that is linked with the Overlay Attribute. Therefore, the TSO/ISPF option 3.3 does not copy the MARKIV module. The PDS data set (...PREP.JCLCNTL) contains two JCL members that run standard IEBCOPY jobs for use in copying the VISION:Builder System Load Library. See members [BLXCOP#1](#) and [BLXCOP#2](#) (in [Appendix A, JCL](#)) for details.

The BLXCOP#1 job deletes the previous version of the copied load library, allocates a new version, and copies the VISION:Builder Target System Load Library to the new user load library.

The BLXCOP#2 job copies the VISION:Builder Target System Load Library to an existing user load library. This is a copy with REPLACE action. The user load library is also compressed in place.

## Step 14 – Customize the Parameter Modules

The VISION:Builder engine and the COMLIB component contain parameters, options, and settings that can be tailored for each installation. The default values are contained in modules that can be modified, assembled, and link edited as needed at any time.

If no modifications are required, skip this step. You can always perform this step at a later time.

The four parameter modules are:

- M4PARAMS    VISION:Builder primary parameter module
- M4SFPARM    VISION:Builder special validation parameter module
- M4LEPARM    VISION:Builder Language Environment Parameter Module
- MARKLIBP    COMLIB Component parameter module

These are listed in detail in [Appendix B, VISION:Builder Parameter Modules](#). The source code for the parameter modules is contained in the Samples data set (...BLSAMP).

## Modifying the Modules

### To modify the modules

1. Create a backup copy of the original source code for future reference.
  2. Once you have reviewed and modified the parameter modules as needed, use the JCL member BLXASM#1 in the PDS data set (...PREP.JCLCNTL) to assemble and link edit the M4PARAMS, M4LEPARM, M4SFPARM, and MARKLIBP modules into the VISION:Builder load library.
- You only need to run the job steps that correspond to the modules being changed.
  - For each step you choose to run, specify the Samples data set (...BLSAMP) name, the source member name, and the load library name.

## Storing the Modules

The M4PARAMS, M4LEPARM, M4SFPARM, and MARKLIBP modules are independently loaded at run time and, therefore, can be stored in load libraries that are separate from VISION:Builder and COMLIB. If used, the separate load library needs to be concatenated before the VISION:Builder and COMLIB load libraries. In this way, you can tailor several different versions of these modules to satisfy varying requirements for separate user groups.

## Step 15 – Install the DB2 Database Access Module MARKSQL

Only customers using the DB2 Database option in VISION:Builder need to perform this step.

### Using Slots to Access DB2 Tables

The MARKSQL module contains slots used to hold the SQL statements needed to access the tables of your DB2 databases during application processing runs. This module must go through the DB2 application program preparation process described in IBM's DATABASE 2™ Application Programming Guide manuals. The MARKSQL source code module supplied in the Samples data set (...BLSAMP) is assembled to generate source statements. These source statements become the plan required by DB2 to run VISION:Builder applications that access DB2 tables.

## Controlling the Number of Statement Slots

The number of statement slots in the generated MARKSQL plan modules limits the number of tables that can be accessed in an application run. In the MARKSQL module, the parameter &MAX controls the number of statement slots generated. The number of statements needed to access DB2 tables in an execution run varies according to which processing options are used in the application. For example, if MOSAIC processing is not used, the number of tables that can be accessed is equal to the &MAX value.

However, if MOSAIC processing is used to access a DB2 table, three statement slots within the MARKSQL plan module are used instead of one. This results in lowering the number of tables available in the run. Applications using the updating facility can also cause multiple statement slots to be used per table. The application source listing shows the total number of statements that are prepared for use during a particular processing run, as well as a description of each prepared statement.

The MARKSQL source module supplied in the Samples data set (...BLSAMP) allows for 100 statement slots. An application program can use up to 100 individual DB2 tables. You can change the parameter within the MARKSQL module to allow for access to more or less than 100 tables (statement slots). To change the parameter, change the value in the following statement that appears after the introductory comments of the MARKSQL module:

```
&MAX SETA 100 MAXIMUM NUMBER OF SQL STATEMENTS PER APPL.
```

## Assembling and Preparing MARKSQL

Once the appropriate parameter value has been set or the default value accepted, the MARKSQL module must be assembled to generate the final source statements to be prepared for DB2. This assembly does not generate any object code but uses the assembler as a source code generator. The source code generated is precompiled (to produce a DBRM), assembled (to produce an object module), and link edited (to produce a load module).

Use the JCL member BLXDB2#1 in the PDS data set (...PREP.JCLCNTL) to assemble and prepare the MARKSQL modules. The MARKSQL module is prepared using the following attach facilities, to correspond with the three methods for attaching to DB2:

■ CALL ■ IMS™ ■ TSO

**Note:** During the link edit step, some linkage editors may issue warning messages and a condition code 4. These can be ignored. For example, message IEW2646W issued by the Binder Linkage Editor is a warning about RMODE Conflicts when the IBM Module DSNALI is included. The module MARKSQLC is correctly linked and the message can be ignored.

The JCL member BLXDB2#1 uses the standard IBM procedure DSNHASM to accomplish the DB2 preparation process and produce three separate modules named MARKSQLC, MARKSQLI, and MARKSQLT. Also, the MEM parameter is used to specify plan names. The module names and default plan names for each attach facility are as follows:

Module Name	Attach Facility	Plan Name
MARKSQLC	CALL	MARKDB2
MARKSQLI	IMS	MARKDLI
MARKSQLT	TSO	MARKIV

You might need to confer with your DB2 database administrator before proceeding with this process. You only need to prepare the MARKSQL modules that correspond to the attach facilities you will use.

When executing your application under one of the attach facilities, the plan name must be provided to DB2.

When using CALL Attach,	supply the plan name in the Run Parameter (RP) DB2 statement within the application source code.
When using IMS Attach,	supply the plan name on the control statement input from the specified DDITV02 DD.
When using TSO Attach,	the terminal monitor program RUN control statement contains the plan name entry.

Refer to the [VISION:Builder for OS/390 Environment Guide](#) for samples of the JCL needed to run your application using the three attach facilities.

### Using the BIND Function

After the required MARKSQL modules are preprocessed, compiled, and link edited, the application plans generated by the DB2 preprocessor must undergo an additional process called binding.

The BIND function can be invoked using either:

- DB2I (DB2 Interactive) under the TSO terminal monitor program with the appropriate control statements
- or
- Batch JCL

**Note:** VISION:Builder does not require repeatable read isolation. Specify cursor stability isolation when binding the plans to allow greater concurrent access to your DB2 tables.

Refer to the *IBM DATABASE2 Application Programming Guide* for your environment for additional information on the bind process. You may need to confer with your DB2 database administrator before proceeding with the bind process. You only need to bind the plan names for the attach facilities that you will be using.

You can choose either of the two methods to perform the BIND process for the prepared MARKSQL modules that will be used to Attach to DB2. [Figure 5-2](#) shows the panels displayed when performing the BIND using the DB2I (DB2 Interactive) facility. Or, you can use the JCL member BLXDB2#2 in the PDS data set (...PREP.JCLCNTL) to bind the plan names in a batch job.

If at any time you make changes to the MARKSQL modules, you need to repeat the preparation and bind process. Be aware that you must use a BIND REPLACE action, not REBIND, when you perform a new BIND.

**Note:** Member name MARKDB2 on the third display is for the CALL Attach. For the IMS Attach, change it to MARKDLI. For the TSO Attach, change it to MARKIV.

```

                                DB2I PRIMARY OPTION MENU                SSID: D61A
COMMAND ==> 5

Select one of the following DB2 functions and press ENTER.

 1 SPUFI                (Process SQL statements)
 2 DCLGEN               (Generate SQL and source language declarations)
 3 PROGRAM PREPARATION  (Prepare a DB2 application program to run)
 4 PRECOMPILE           (Invoke DB2 precompiler)
 5 BIND/REBIND/FREE     (BIND, REBIND, or FREE plans or packages)
 6 RUN                  (RUN an SQL program)
 7 DB2 COMMANDS         (Issue DB2 commands)
 8 UTILITIES            (Invoke DB2 utilities)
 D DB2I DEFAULTS       (Set global parameters)
 X EXIT                 (Leave DB2I)

PRESS:                  END to exit          HELP for more information

```

```

                                BIND/REBIND/FREE                      SSID: D61A
COMMAND ==> 1

Select one of the following and press ENTER:

 1 BIND PLAN            (Add or replace an application plan)
 2 REBIND PLAN          (Rebind existing application plan or plans)
 3 FREE PLAN            (Erase application plan or plans)
 4 BIND PACKAGE         (Add or replace a package)
 5 REBIND PACKAGE       (Rebind existing package or packages)
 6 REBIND TRIGGER PACKAGE (Rebind existing trigger package or packages)
 7 FREE PACKAGE         (Erase a package or packages)

PRESS:  ENTER to process  END to exit          HELP for more information

```

```

                                BIND PLAN                          SSID: D61A
COMMAND ==>

Enter DBRM data set name(s):
 1 MEMBER ..... ==> MARKDB2
 2 PASSWORD ..... ==>
 3 LIBRARY ..... ==> PUBLIC.DBRMLIB.DATA
 4 ADDITIONAL DBRMS? ..... ==> NO        (YES to include more DBRMs)

Enter options as desired:
 5 PLAN NAME ..... ==> MARKDB2        (Required to create a plan)
 6 CHANGE CURRENT DEFAULTS? .. ==> NO  (NO or YES)
 7 ENABLE/DISABLE CONNECTIONS? ==> NO  (NO or YES)
 8 INCLUDE PACKAGE LIST?..... ==> NO  (NO or YES)
 9 OWNER OF PLAN (AUTHID)..... ==>    (Leave blank for your primaryID)
10 QUALIFIER ..... ==>                (For tables, views, and aliases)
11 CACHESIZE ..... ==>                (Blank, or value 0-4096)
12 ACTION ON PLAN ..... ==> REPLACE   (REPLACE or ADD)
13 RETAIN EXECUTION AUTHORITY. ==> YES (YES to retain user list)
14 CURRENT SERVER ..... ==>          (Location name)
15 INCLUDE PATH?..... ==> NO          (NO or YES)

PRESS:  ENTER to process  END to save and exit  HELP for more information

```

Figure 5-2 Using DB2I to Perform the BIND Function

## Teradata Information

Customer sites using the Teradata Database System need to prepare their MARKSQL module as follows:

1. To install the MARKSQL module for use with Teradata databases, use the JCL member BLXDB2#T in the PDS data set (...PREP.JCLCNTL) to assemble and prepare the MARKSQL module. The result is the module named MARKSQLT.

**Note:** This module name is the same as the TSO Attach module used for access to IBM DB2 tables. You may want to store the Teradata version of MARKSQLT in a separate load library, especially if your site uses both IBM and Teradata databases.

2. Once the module is prepared, you can perform any Teradata tasks needed to get the module ready for run-time use (such as binding).
3. When running your VISION:Builder application, you must provide the appropriate DD statements needed by Teradata for profile and control information. Refer to your Teradata manuals for the proper coding of this information.
4. The Teradata run-time library must be included in the STEPLIB, JOBLIB, or link-list concatenation when running your applications.
5. The VISION:Builder application source code MUST NOT contain a Run Parameter (RP) DB2 statement because this causes VISION:Builder to attempt a Call Attach using the MARKSQLC module. The absence of an RP DB2 statement when running a standard (non-IMS) VISION:Builder processing step, causes an Attach using the MARKSQLT module.

## Step 16 – Install the PAL File Definitions and Requests

If you will be using the Program Analyzer (PAL) facility, you need to perform this step. The VISION:Builder file definitions and application requests are cataloged into a common library (M4LIB) for use in the production of the various PAL Reports.

For proper PAL support, three file definitions (IGCPALVB, IGCPALWK, and IGCPALRS) and a request group (IGCPAL) must be cataloged. The request group IGCPAL contains the following request names:

IGCPALFT	IGCPALDR	IGCPAL2P	IGCPAL4	IGCPAL5P
IGCPALRS	IGCPAL1	IGCPAL3	IGCPAL4P	IGCPAL6
IGCPALEF	IGCPAL2	IGCPAL3P	IGCPAL5	IGCPAL7

To install the PAL file definitions and requests, you must initialize an M4LIB and catalog the file definitions, the individual requests, and the request group into the M4LIB. Use the JCL member BLXPAL#1 in the PDS data set (...PREP.JCLCNTL) to catalog the PAL items. The job catalogs the items into a BDAM format M4LIB.

The INIT (initialize an M4LIB) and DEFRUN1 (definition run) steps should receive a condition code 0, indicating successful completion. The PROCRUN (processing scan only run) step receives a condition code 4, indicating successful completion of the scan only run. Review the source listing to ensure that no type 3 or higher error messages were issued.

VISION:Builder uses this M4LIB data set when the PAL facility is used to produce reports.

## Step 17 – Relink Static Own Code Integration

This step is only needed if you use the Static Integration Facility for your user-written M4OWN module. The purpose and use of this facility is described in the Environment Manual. This step can be skipped if static own code integration is not used.

Use JCL member BLXRLK#1 in the PDS data set (...PREP.JCLCNTL) to run a job that relinks the VISION:Builder main program module and includes your user-written version of the module M4OWN. You need to supply your M4OWN module as an object or a load module. See the JCL comments that indicate the data sets needed for the link edit job.

### Error Messages to Ignore

During the link edit, various warning and error messages are issued. The following messages can be safely ignored:

```
IEW0461  WARNING-SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS  
         SPECIFIED.  
  
IEW0161  WARNING-EXCLUSIVE CALL FROM SEGMENT NUMBER PRINTED TO SYMBOL PRINTED  
         - XCAL WAS SPECIFIED.
```

These messages cause a condition code 4, which can be safely ignored.

Also, if you comment out the DLILIB DD (or the OBJLIB DD) statement in the link edit JCL, a condition code 8 occurs and the following message is issued and can be ignored:

```
IEW0432  ERROR-LIBRARY NAME PRINTED CANNOT BE OPENED, DD STATEMENT MAY BE  
         MISSING.
```

Link edit storage requirements vary from installation to installation. However, the normal region size and default link edit size parameter values used at your installation for links to a loadlib should be suitable for this link edit step. If you have a problem, you can use a region size of 2 MB and a link edit size parameter of (310K, 84K) as a guideline.

**Note:** If you need to restore the original M4OWN module delivered with the system to turn off and remove Static Own Code, the PDS data set (...PREP.JCLCNTL) contains the original M4OWN object code. Point to the PDS data set (...PREP.JCLCNTL) on the OBJLIB DD statement and run the BLRLNK job.

## Step 18 – Set Up for Use with the TSO Command Processor

The following sections are for sites that use the OLX, OQL, OFI, BQL, and BFI features of VISION:Builder. If you do not use any of these facilities, proceed to [Step 19 – Copy VISION:Builder Message Modules to LPA on page 5-14](#). These features continue to be delivered as part of the system for compatibility purposes and in support of legacy systems still using these features of the product.

### OQL and BQL Parameter Module Modification

The Online Executive (OLX) facility uses modules containing parameters, options, and settings that can be tailored to each installation. The default values are contained in modules that can be modified, assembled, and link edited as needed at any time. If no modifications are required, you can skip this section.

The two parameter modules are:

- OQLPARM    Online Query Language (OQL, OLX, OFI) parameter module
- BQLPARM    Batch Query Language (BQL, BFI) parameter module

These are listed in detail in [Appendix B, VISION:Builder Parameter Modules](#). The source code for the parameter modules is contained in the Samples data set (...BLSAMP).

If you are going to modify any of these modules, create a backup copy of the original source code.

Once you have reviewed and modified the parameter modules as needed, you can use the JCL member BLXASM#2 in the PDS data set (...PREP.JCLCNTL) to assemble and link edit new versions of the modules into the VISION:Builder component load libraries. The JCL in member BLXASM#2 has examples for the assembly and link edit of both of the parameter modules. You only need to run the steps that correspond to the modules you change.

For each step you choose to run, specify the Samples data set (...BLSAMP) name, the source member name, and the load library name.

### TSO Help Data Set

If your site does not utilize the OLX, OFI, or OQL facility, this section can be skipped.

The TSO help members for the OLX, OFI, or OQL facility must be made available to the general TSO help processor during terminal sessions.

The TSO help members can be copied to a separate help data set that is concatenated with the other data sets specified on the SYSHELP DD statement in the TSO logon procedure. Another option is to copy the help members directly into one of the help data sets already specified on the SYSHELP DD statement in the TSO logon procedure.

Use the JCL member BLXOLX#1 in the Samples data set (...BLSAMP) to copy the TSO help members to a help data set. The BLXOLX#1 job allocates a new data set and copies the help members from the PDS data set (...PREP.JCLCNTL) to the new data set. If you are using an existing data set, skip the ALLOC step in the JCL. The COPY job replaces members of the same name in the existing data set.

### OLX Command Processors

Users of the OLX facility may want to copy some of the command processing modules to the SYS1.LINKLIB (or concatenation thereof) so that TSO can find them. This allows users to specify the VISION:Builder load library as an operand of the M4EXEC command. Otherwise, the VISION:Builder load library must be part of the STEPLIB DD or allocated as a STEPLIB for the TSO session. This specification is optional.

Use the JCL member BLXOLX#2 in the PDS data set (...PREP.JCLCNTL) to copy some of the OLX Command Modules to a SYS1.LINKLIB or concatenation thereof. Specify the VISION:Builder load library and the SYS1.LINKLIB data set names. The COPY job replaces members of the same name in the copied to data set.

## Step 19 – Copy VISION:Builder Message Modules to LPA

This optional step can be done if you want to copy the VISION:Builder message modules to your LPA libraries for shared access.

The VISION:Builder messages are contained in modules within the VISION:Builder load library. These modules are loaded into memory (the region) as needed during the various execution runs. The message modules are marked as reentrant and could be placed in the LPA. They are then shared by all VISION:Builder runs, which reduces the amount of storage used in the region for each job. The size of each message module is 4K. An index and global message module are also used in the message handling mechanism.

Use the JCL member BLXMSG#1 in the PDS data set (...PREP.JCLCNTL) to copy the VISION:Builder message modules from the VISION:Builder load library to an LPA load library. Specify the VISION:Builder load library and the LPA load library data set names. The COPY job replaces members of the same name in the copied to data set.

## Step 20 – Install VISION:Workbench for DOS

One of the components of the VISION:Builder software system is VISION:Workbench™ for DOS. It is a PC-based application programming development tool. This component runs on a PC workstation under DOS. It can also be run in a DOS window under Windows 95®, Windows 98®, and Windows NT®.

This tool helps VISION:Builder users to prepare their definitions and applications. All coding is checked for errors and inconsistencies at the PC without the need to connect to the host. Once the application and definitions are ready, the users transfer the VISION:Builder source statements to the host for actual submission and processing.

VISION:Workbench for DOS is delivered on compact disc. The CD should be distributed among all VISION:Builder users. The CD is NOT copy protected, you can make unlimited copies as needed. Contact Computer Associates Technical Support if you cannot locate the CD in your installation package or have other problems. See [Contacting Computer Associates on page 1-11](#) for more information.

Information about installing and using VISION:Workbench for DOS is contained in the VISION:Workbench for DOS User's Guide.

## Step 21 – Set Up VISION:Workbench for ISPF Requirements

This section contains the following sections:

- [Allocating VISION:Workbench for ISPF Run-Time Libraries on page 5-16](#)
- [Allocation Requirements on page 5-16](#)
- [Using STEPLIB and System Link Library on page 5-19](#)
- [Using the LIBDEF Feature on page 5-19](#)
- [More about ISPF FILE Allocations on page 5-19](#)
- [List Data Set and Internal Work Files on page 5-20](#)

## Allocating VISION:Workbench for ISPF Run-Time Libraries

VISION:Workbench for ISPF runs as an application under IBM's ISPF/PDF Facility, which is an extension of TSO, and takes advantage of the many standard services available under ISPF.

To integrate VISION:Workbench for ISPF into the ISPF environment, the VISION:Workbench for ISPF libraries and the appropriate VISION:Builder, VISION:Transact™, VISION:Inform™, and COMLIB component load libraries must be made available to the ISPF facility.

The primary methods for making the VISION:Workbench for ISPF libraries known to ISPF are as follows:

- Add the libraries to the TSO logon procedure or the ISPF “start-up” CLIST.
- Use the ISPF LIBDEF service to dynamically modify your ISPF library concatenations.

Check with the systems group at your facility to confirm which method you should use to get the VISION:Workbench for ISPF libraries allocated for ISPF sessions.

### To allocate the necessary VISION:Workbench for ISPF libraries

1. Determine how your existing ISPF libraries are being allocated.
2. Concatenate the VISION:Workbench for ISPF libraries to the existing ISPF library allocations.
3. Allocate the VISION:Workbench for ISPF libraries in front of each concatenation sequence.

## Allocation Requirements

The following list shows the required ISPF ddnames and the VISION:Workbench for ISPF data sets that should be associated with them. All VISION:Workbench for ISPF data set names displayed are the suggested names shown earlier in this document. Change these names to reflect those names actually used during your installation process.

ddname:           SYSPROC

Data set name:   BUILDER.R140.SMPE.T.WBCLIST

ddname:           ISPLLIB

Data set names:   BUILDER.R140.SMPE.T.BLSYSL  
                    TRANSACT.TR075.GENLIB (VISION:Transact sites)  
                    INFORM40.LOADLIB (VISION:Inform sites)

**Note:** The VISION:Workbench for ISPF CLIST library was delivered on the installation tape as a fixed blocked data set with a record length of 80. If your installation prefers a variable blocked format, you may want to copy the contents of this library over to a different CLIST library of the proper format for your site.

ISPLLIB functions as a task library. It is searched before the STEPLIB allocations, system link libraries, or the system link pack area.

VISION:Workbench for ISPF Release 6.0 requires the specific releases of the associated software products and components, as shown in the following table.

Software Product	Release Number
VISION:Builder	Release 14.0
VISION:Transact	Release 7.5
VISION:Inform	Release 4.0
COMLIB	Release 4.5

Additionally, with VISION:Builder Release 14.0 and VISION:Transact Release 7.5, the IBM Language Environment (LE, formerly LE/370) is utilized. The LE run-time library must be available when running the validation function of VISION:Workbench for ISPF. If the LE modules are not available at your facility, RSMs (Restricted System Modifications) are available that, when installed in the VISION:Builder and VISION:Transact load libraries, causes VISION:Builder and VISION:Transact to bypass using any LE modules.

## Panel Library

ddname: ISPPLIB  
Data set name: BUILDER.R140.SMPE.T.WBPANEL

If you have chosen to preprocess your VISION:Workbench for ISPF panel library, concatenate the preprocessed panel library, rather than the panel source library, to this ddname. Preprocessing the panel library is an optional installation step discussed in [Optional Setup for VISION:Workbench for ISPF on page 5-27](#) of this document.

## Locate Text

ddname: ISPMLIB  
Data set name: BUILDER.R140.SMPE.T.WBMSG

This ddname is used by ISPF to locate the text of all messages issued by ISPF applications like VISION:Workbench for ISPF.

### Locate Skeletons

ddname: ISPSLIB  
Data set name: BUILDER.R140.SMPE.T.WBSKELS

The ISPSLIB ddname is used to specify the location of ISPF file tailoring skeletons used by VISION:Workbench for ISPF.

### Generate Facility

ddname: ISPFIL  
Data set name: This data set name should reference a file tailoring output library.

Sites that use the VISION:Workbench for ISPF “Generate” facility must preallocate this file.

### IMPORT Option

The IMPORT option of VISION:Workbench for ISPF requires a data set to save information entered during the IMPORT function dialogs. The entered information is then available from session to session. The data set should be preallocated and cataloged. The data set characteristics are as follows:

- DSORG: PO
- RECFM: FB
- LRECL: 80
- BLKSIZE: multiple of 80
- SPACE: (TRK,(5,2,2))

### Start-up CLIST

Once the data set has been defined, you need to add the data set to the ISPF start-up CLIST allocations. The free and allocation entries for the ddname DEFTLIB should be coded as follows:

- FREE: F(DEFTLIB)
- ALLOC: F(DEFTLIB) DA('user.defined.name') SHR

To allow for automatic dynamic allocation and cataloging of the data set for each unique user, you could insert the following sample CLIST statements into the startup ISPF CLIST:

```
FREE          F (DEFTLIB)

IF &SYSDSN('BUILDER.R140.&SYSUID..TLIB') = OK THEN +
  ALLOC F(DEFTLIB) DA('BUILDER.R140.&SYSUID..TLIB') SHR
ELSE +
  ALLOC F(DEFTLIB) DA('BUILDER.R140.&SYSUID..TLIB') +
  NEW CATALOG UNIT(SYSDA) SPACE(5,2) DIR(2) +
  DSORG(PO) RECFM(F B) LRECL(80) BLKSIZE(3120)
```

The &SYSUID element entry is replaced by the current user ID when the startup CLIST is activated.

[Appendix C, Sample ISPF Startup CLIST](#) contains a sample ISPF start-up CLIST that shows how the VISION:Workbench for ISPF library allocations can be accomplished.

## Using STEPLIB and System Link Library

Rather than allocating them to ISPLLIB, you can make your VISION:Workbench for ISPF load library and related load libraries available to ISPF using a STEPLIB allocation or system link library allocations. Alternatively, because VISION:Workbench for ISPF is reentrant, you can place its load modules in the system link pack area. VISION:Builder, VISION:Transact, and COMLIB are not reentrant and should not be run from the system link pack area.

## Using the LIBDEF Feature

This feature allows you to dynamically modify your ISPF library concatenations based on the ISPF application you are running.

Use the LIBDEF service to allocate your VISION:Workbench for ISPF CLIST, panel, message, and skeleton libraries.

Do not use this feature to allocate load libraries or the file tailoring output data set for VISION:Builder application generation. VISION:Workbench for ISPF relies on OS/390 (MVS) services to find load modules and to obtain the data set name for file tailoring output. OS/390 (MVS) services do not recognize allocations done using the ISPF LIBDEF service. VISION:Workbench for ISPF does not function properly if LIBDEF is used for ISPLLIB or ISPFIL allocations.

For more information on the LIBDEF service, refer to IBM's ISPF Dialogue Management Guide and Reference manual.

## More about ISPFIL Allocations

The ISPFIL allocation is only applicable to VISION:Builder customers using VISION:Workbench for ISPF. If used, the ISPFIL allocation must not specify a concatenated sequence of data sets.

During the VISION:Workbench for ISPF VISION:Builder job submission process, if you specify the “keep” or “keep/submit” processing option, the generated JCL or CLIST is written to the data set allocated to ISPF. This data set is often referred to as the *file tailoring output* data set.

As mentioned earlier, VISION:Builder users must preallocate this file if they will be using the generate facility. This preallocation must not be done using the ISPF LIBDEF service (see [Using the LIBDEF Feature on page 5-19](#)).

VISION:Transact sites do not need to preallocate ISPF. VISION:Workbench for ISPF dynamically allocates the file tailoring output data set as required and uses its own ddname (M9FTOUT) for this purpose. VISION:Workbench for ISPF does not deallocate any existing ISPF assignments.

The file tailoring output data set must be a partitioned data set. Create this data set with the following recommended characteristics:

- DSORG: PO
- RECFM: FB
- LRECL: 80
- BLKSIZE: any multiple of 80

Normally, each user wants to have a personal file tailoring output data set. This can be accomplished by using the user ID as one of the qualifiers when allocating the data set name. For example, the data set name could be BUILDER.R140.&SYSUID.FTOUTPUT.

[Appendix C, Sample ISPF Startup CLIST](#) contains an example to show how a file tailoring output data set can be allocated in your ISPF start-up CLIST. For more information about allocating your file tailoring output data set, refer to IBM's ISPF Dialogue Management Guide and Reference manual.

## List Data Set and Internal Work Files

### The List Data Set

VISION:Workbench for ISPF uses a list data set that works similarly to the ISPF list data set.

You can preallocate this data set prior to invoking your VISION:Workbench for ISPF session, but preallocation is not required. If this data set has not been preallocated, it will dynamically allocate with a disposition of new when the data set is needed.

If this data set is preallocated, it must have the following characteristics:

- DDNAME: M9LIST
- DSORG: PS or SYSOUT

- RECFM: FBA
- LRECL: 133
- BLKSIZE: any multiple of 133

The list data set corresponds to the ISPF list data set in purpose and function. It is used to hold any output that you request while in VISION:Workbench for ISPF. For example, if you use the utilities to perform a source statement retrieval and you request a hard copy of the source, the source listing is written to this data set.

If you preallocate this data set, no termination processing is attempted at the end of the VISION:Workbench for ISPF session.

If VISION:Workbench for ISPF allocates this data set, a Process List Data Set panel appears during termination processing. This panel functions in the same manner as the ISPF Process List Data Set panel.

You can set up default processing parameters for this data set using the VISION:Workbench for ISPF Parameters selection.

If the list data set is dynamically allocated by VISION:Workbench for ISPF, the naming convention used is &SYSPREF.(&SYSUID.).M9TEMPn.LIST. The &SYSUID qualifier is only used if it differs from the &SYSPREF system prefix.

## Internal Work Files

A VISION:Workbench for ISPF session can use up to five internal work files. These data sets are allocated as needed to the following ddnames:

- M9LST1   ■ M9LST2   ■ M9LST3   ■ M9LST4   ■ M5LIST

M9LST3 and M9LST4 are only required when a 3290 terminal is in use. M5LIST is only used by the VISION:Transact development facility.

These data sets are allocated and deleted as necessary and cannot be preallocated.

You can control some of the dynamic allocation parameters used by VISION:Workbench for ISPF when allocating these data sets by modifying the following two panels in your panel library:

- M9DATPMI - This panel allows you to customize some of the allocation parameters used to allocate the M9LSTn data sets.
- M9DATPMV - This panel allows you to customize some of the allocation parameters used to allocate the M5LIST data set.

These panels allow you to specify unit and space allocations for the internal work files. Just prior to dynamically allocating any of these data sets, VISION:Workbench for ISPF retrieves and uses the allocation information from the appropriate panel variables.

Remember that the units specified on these panels must be known to TSO and must be eligible to contain permanent data sets.

The naming convention used by VISION:Workbench for ISPF is as follows

Data Set	Naming Convention
M9LSTn	&SYSPREF.(&SYSUID.).M9TEMPn.LSTn
M5LIST	&SYSPREF.(&SYSUID.).M9TEMPn.M5LIST

In both cases, only use the &SYSUID qualifier if it differs from the &SYSPREF system prefix.

## Invoking VISION:Workbench for ISPF

VISION:Workbench for ISPF is designed to support VISION:Builder, VISION:Transact, and VISION:Inform, along with the shared COMLIB component. Although these three products and the shared component have a commonality of elements and specifications, there are also separate elements that are unique to each product. VISION:Workbench for ISPF has a common entry point that provides for the selection of a subsection appropriate for each specific product. Full functionality of each subsection of VISION:Workbench for ISPF is dependent upon the presence of the separate product software. At least one product, along with the COMLIB component, must be available for VISION:Workbench for ISPF to function.

VISION:Workbench for ISPF runs under IBM's ISPF/PDF facility, which you start in one of the following ways:

- By executing an ISPF "start-up" CLIST once you are logged on to TSO.
- From the TSO logon procedure.

In either case, the appropriate ISPF data sets are allocated and a menu panel appears for the user to make a selection and activate the wanted services or dialogs.

Use one of the following methods for invoking ISPF dialog applications:

- [Adding an Option to a Standard ISPF Primary Menu on page 5-23](#). You can add a selection option to the standard ISPF primary menu panel used at your facility. Users can then select the designated option to invoke

VISION:Workbench for ISPF. When invoked, a VISION:Workbench for ISPF entry menu appears and the user selects the appropriate option for entry to the wanted subsection application (VISION:Builder, VISION:Transact, and so on).

- [Using the ISPSTART Command on page 5-25](#). You can create a CLIST that allocates the appropriate ISPF data sets and uses the ISPSTART command to directly invoke VISION:Workbench for ISPF. This allows users to invoke VISION:Workbench for ISPF without going through the standard ISPF primary menu. The standard IBM ISPF options would not be available for selection by the user when this method is used.

There are variations of these basic methods that can be used to accomplish the same results. Your IS staff knows which method works best for your facility. Whichever method is used, the proper ISPF environment and appropriate ISPF data sets, along with the VISION:Workbench for ISPF and associated product data sets, must be established and allocated for everything to function properly.

## Adding an Option to a Standard ISPF Primary Menu

In [Appendix D, Invocation Panels](#), the [XSR@PRIM](#) panel shows a sample of an ISPF primary menu panel specification. The arrows in the figure point to the specifications that can be added to cause VISION:Workbench for ISPF to be invoked. You only need to add the specifications that are appropriate for your facility.

To add a selection option to the standard ISPF primary menu panel

1. Add an assignment to the INIT section of the panel, as shown in the following statement:

```
&M9PRODUCT = 'Workbench'
```

This assignment sets an internal variable in VISION:Workbench for ISPF that is used in some of the panel displays.

2. To invoke the VISION:Workbench for ISPF Primary Selection Menu, add the WB specification to your primary menu. The specification adds an option code (WB) to the panel display (top portion) and an action (PROC) to be taken when the option is selected.
3. The system displays the following line:

```
% WB +WORKBENCH          -%VISION:Workbench Facility - Release 6.0
```

The % specifies to highlight the following text and the + specifies to use normal intensity for the following text. These are standard ISPF attribute indicators. The remaining text and characters are display-only and therefore, can be almost anything you choose.

4. The following line causes ISPF to display the panel M9PRIM when the option WB is entered on the ISPF primary menu option line:

```
WB,' PANEL (M9PRIM)' /* Invokes VISION:Workbench Selection Menu */
```

The line must be keyed in uppercase. The /\* ....\*/ is just a comment and can be ignored or used as a reference.

- The ISPF WB action causes the M9PRIM panel to appear and a transfer control to the actions specified on that panel from subsequent user interaction.
  - The M9PRIM panel is the primary selection menu for VISION:Workbench for ISPF. [M9PRIM on page D-2](#) shows the M9PRIM panel specifications.
5. From the VISION:Workbench for ISPF primary selection menu, enter the appropriate option to invoke the wanted subsection.

You can choose to bypass the VISION:Workbench for ISPF primary selection menu and go directly to the desired subsection by adding some or all of the remaining specifications shown by the arrows in [XSR@PRIM on page D-1](#).

6. The system displays the following lines:

```
% BL   +Builder           -%VISION:Builder 14.0 Workbench
% TR   +Transact          -%VISION:Transact 7.5 Workbench
% IN   +Inform            -%VISION:Inform 4.0 Workbench
```

The % specifies to highlight the following text and the + specifies to use normal intensity for the following text. These are standard ISPF attribute indicators. The remaining text and characters are display-only and therefore, can be almost anything you choose. The purpose is to instruct the user to use the BL, TR, and IN characters as an option on the option line and cause the action specified in the PROC section of the panel coding.

7. The following PROC section lines cause ISPF to transfer control to the program M9BOOT when the option BL, TR, or IN is entered on the ISPF primary menu option line:

```
BL,' PGM (M9BOOT) PARM (BDM4) NOCHECK'
TR,' PGM (M9BOOT) PARM (ODM5) NOCHECK'
IN,' PGM (M9BOOT) PARM (PMM4) NOCHECK'
```

The line must be keyed in uppercase. The PARM values are passed to the M9BOOT program and causes the appropriate subsection selection menu to appear.

8. You need to add one final specification to the PROC section of the panel, as shown in the following statement:

```
&GVNXTSEL = .TRAIL
```

This assignment sets an internal variable in VISION:Workbench for ISPF such that any trailing command options are available for processing.

Only use the options and actions for the products that are appropriate for your facility.

If you do not have all the companion product software, see [Using Other VISION:Workbench for ISPF Subsections on page 5-26](#). There is information on how to use the other portions of VISION:Workbench for ISPF without actually having the companion product software.

The above specifications for panels are standard coding as provided by IBM for their ISPF/PDF environment. Check with your systems people if you are not sure what is the best method for your facility.

## Using the ISPSTART Command

VISION:Workbench for ISPF can be directly invoked from TSO using the ISPSTART command. You can write a CLIST that allocates all the ISPF data sets, along with the VISION:Workbench for ISPF data sets, and then executes the ISPSTART command to invoke VISION:Workbench for ISPF directly. The CLIST is almost identical to the standard ISPF start-up CLIST. Use this method to bypass the standard ISPF primary menu. The options normally available to ISPF from the primary menu could not be accessed.

[Appendix C, Sample ISPF Startup CLIST](#) shows a sample ISPF start-up CLIST. The ISPSTART command appears at the end of the CLIST and can be changed to invoke the VISION:Workbench for ISPF selection menu (M9PRIM) or the entry program (M9BOOT) directly.

To invoke VISION:Workbench for ISPF using the ISPSTART command:

1. Invoke the VISION:Workbench for ISPF selection menu using the following statement:
2. Invoke a VISION:Workbench for ISPF subsection menu using the following statement:

```
ISPSTART PANEL(M9PRIM)
```

```
ISPSTART PGM(M9BOOT) PARM(yyyy)
```

where:     yyyy     identifies the product subsection, as follows:

- BDM4 - VISION:Builder 14.0 Workbench
- ODM5 - VISION:Transact 7.5 Workbench
- PMM4 - VISION:Inform 4.0 Workbench

**Note:** Only one subsection can be started per TSO session.

These specifications for CLISTs are standard coding as provided by IBM for their TSO environment. Check with your systems people if you are not sure what is the best method for your facility.

## Using Other VISION:Workbench for ISPF Subsections

VISION:Workbench for ISPF is designed to be fully-functional when the companion product software is also installed and available to ISPF. However, VISION:Workbench for ISPF can also be used when the companion product software is not installed. One of the software products must be available so that, as a minimum, the COMLIB component is present and available to VISION:Workbench for ISPF.

If your facility already has all the companion product software, VISION:Builder, VISION:Transact, VISION:Inform, or if you are not interested in exploring the other portions of VISION:Workbench for ISPF, then you can skip this section.

When the associated software product is not available, application validation does not function, but the data entry edits are still active. This gives users a chance to explore the other portions of VISION:Workbench for ISPF.

Change the PARM(xxxx) in the following locations when the associated product software is not installed and available:

- Panel M9PRIM (described in [Appendix D, Invocation Panels](#))
- The tailored ISPF primary menu panel
- The ISPSTART command specifications

The following table shows the before and after changes for the PARM entry.

Before	After	Comment
PARM (BDM4)	PARM (BD )	replace the M4 with two blank spaces to indicate that VISION:Builder is not available
PARM (ODM5)	PARM (OD )	replace the M5 with two blank spaces to indicate that VISION:Transact is not available
PARM (PMM4)	PARM (PM )	replace the M4 with two blank spaces to indicate that VISION:Inform is not available

After you have made these changes, perform the following steps to ensure that a product parameters module is available to make the other portions of VISION:Workbench for ISPF function:

- Use the two default parameter modules in the VISION:Builder system load library (BUILDER.R140.SMPE.T.BLSYSL).
- Rename these modules, depending on which other portion you want to be functional.
  - For VISION:Builder and VISION:Inform, rename the M4PRMMOD to M4PARAMS.
  - For VISION:Transact, rename the M5PRMMOD to FIVEPARM.

Once the appropriate specifications have been adjusted, VISION:Workbench for ISPF will function in the subsections even if the companion software product is not installed. Remember, the application validation will not function, but the data entry edits are still active.

## Optional Setup for VISION:Workbench for ISPF

### Preprocessing Your Panel Library

ISPF offers a panel preprocessing utility called ISPPREP. This utility can be used to convert your VISION:Workbench for ISPF panels into an encoded format that significantly improves panel display performance while using VISION:Workbench for ISPF. A preprocessed panel library also takes up to 20% less space than an unprocessed panel library.

Once a panel has been preprocessed and is in an encoded display format, it cannot be modified. To change a preprocessed panel, you must modify the original panel source member and rerun ISPPREP for that panel.

Not all panels can be preprocessed. There are restrictions that prevent ISPPREP from successfully converting certain panels. The following VISION:Workbench for ISPF panels are bypassed automatically when ISPPREP is run:

- M9HCAPBR     ■ M9HCAPPF     ■ M9SVAPBR     ■ M9TBAPTP
- M9HCAPDA     ■ M9HCAPSF     ■ M9SVAPPM
- M9HCAPED     ■ M9HCAPSS     ■ M9TBAPTB

To preprocess your VISION:Workbench for ISPF panels:

1. Allocate a second panel library to hold the preprocessed panels. Leave your original panel source library unchanged.

To preprocess your panel library:

**Note:** If you do not want to preprocess your panel library, skip to [Customizing Job Submission Skeletons on page 5-29](#).

1. *Allocate an ISPF log data set*

Ensure that you have an ISPF log data set allocated. The preprocess utility writes information messages to this data set.

2. *Allocate a new panel library*

Allocate this panel library with the same characteristics as your VISION:Workbench for ISPF panel source library. The space allocation can be reduced to 80 primary tracks. The directory blocks can be reduced to 70 tracks unless you set the Save Statistics option to Yes on the ISPPREP panel, in which case you must increase the directory blocks to 225.

3. *Run the preprocess utility*

- Go into ISPF and select the Command option (option 6) from the primary menu.
- Enter the command ISPPREP on the TSO command line. A preprocess utility panel appears.
- In this panel, specify the source (unprocessed) data set and the target data set where the processed panels will be stored. Type in the appropriate information to convert all panels and press Enter.

The time required to complete the panel conversion process varies from installation to installation. At the Computer Associates installation, ISPPREP takes about 8 minutes to complete. Informational messages appear during this time to tell you how many panels have been processed.

Do not worry about the panels that cannot be preprocessed at this time. The ISPPREP utility will recognize that these panels cannot be encoded and will automatically skip them.

- When this process is finished, view or print your ISPF log. You can view the log using ISPF option 7.5–Dialog Test, Log option. The log contains informational messages from the conversion process that pertain to the panels that could not be converted.

4. *Copy the following unprocessed panels:*

- M9HCAPBR    ■ M9HCAPPF    ■ M9SVAPBR    ■ M9TBAPTP
- M9HCAPDA    ■ M9HCAPSF    ■ M9SVAPPM
- M9HCAPED    ■ M9HCAPSS    ■ M9TBAPTB

After the conversion process is complete, use the ISPF Copy utility (option 3.3) to copy the unprocessed panels listed above from your VISION:Workbench for ISPF panel source library to your new preprocessed panel library.

5. *Allocate the new library to ISPLIB*

The preprocessed panel library now contains all of your VISION:Workbench for ISPF panels. Allocate this library to the ISPF ddname ISPLIB so that ISPF uses the encoded VISION:Workbench for ISPF panels rather than the source versions. You can remove your VISION:Workbench for ISPF panel source library from the ISPLIB concatenation.

6. *Modify Panels After Preprocessing*

If you change a VISION:Workbench for ISPF panel, you must edit it in its source format. Once the modification is complete, replace it in the preprocessed panel library by running the panel through the ISPPREP utility.

## Customizing Job Submission Skeletons

There are four default file tailoring skeletons, accompanied by four default user panels (see [Appendix E, Skeleton and User Panel Listings](#) for more information). These are used by the VISION:Workbench for ISPF “generate” subsystems for VISION:Builder and VISION:Transact. These skeletons and panels are as follows:

Panel M9BGUPNL and Skeleton M9BGTS	This panel/skeleton combination is used to submit VISION:Builder background jobs.
Panel M9FGUPNL and Skeleton M9FGTS	This panel/skeleton combination is used to execute VISION:Builder foreground jobs.
Panel M9GCTPU2 and Skeleton M9GCTSBG	This panel/skeleton combination is used to submit VISION:Transact background jobs.
Panel M9GCTPU1 and Skeleton M9GCTSFG	This panel/skeleton combination is used to execute VISION:Transact foreground jobs.

To customize job submission skeletons:

1. As a safety measure, create a backup before you begin to modify any of the panels or skeletons.
2. Before you actually start customizing your VISION:Workbench for ISPF file tailoring skeletons, run a few job submission tests using the default versions of the skeletons and user panels. When making these test runs, specify a processing option of “Keep” on the file tailoring option panel. This causes your submission JCL or CLIST to be written to the file tailoring output data set, but the job does not execute.
3. Review the generated JCL and CLISTs to:
  - become better acquainted with how the file tailoring skeletons work.
  - see exactly where you need to make changes to conform to your installation standards.
4. If you are currently using VISION:Workbench for ISPF job submission skeletons from a previous release, you can continue to use them with this release. You may have to change data set names to reflect the new release library names, but no other modifications should be necessary.

If you do not have working skeletons from a previous VISION:Workbench for ISPF release, use the skeletons in this release as a starting point and modify them so that they work properly for your site.

5. The sample file tailoring skeletons distributed with the system refer to variables from the corresponding sample user panels. To use a sample skeleton, you must specify during job generation which sample user panel should appear. Once it appears, you should complete all the entries.

The use of user panels is not mandatory. This feature makes the system more flexible, but if you would rather bypass the user panel, you can. Instead, just hard code the user panel information in your job submission skeletons.

For more information about file tailoring and file tailoring skeletons, refer to IBM's ISPF Dialogue Management Guide and Reference manual.

## LMF SUPPORT

If your installation uses the Library Management Facility (LMF), a VISION:Workbench for ISPF APAR, previously a restricted system modification (RSM), is available that places an LMF lock on any members being edited from an LMF-controlled library. The APARs (RSMs) are described in [Step 13 – APPLY Customizing APARs on page 5-1](#). For more information, contact Computer Associates Technical Support (see [Contacting Computer Associates on page 1-11](#)).

## Step 22 – Quick Start Utility Setup

**Note:** Refer to the [VISION:Builder for OS/390 Getting Started Guide](#) for information about quick start utilities.

The VISION:Builder System contains four utilities to quick start the user in developing file definitions. These utilities convert existing COBOL, DB2, VISION:Inquiry, and VISION:Results table and file definitions into a VISION:Builder file definition that is then tailored for use. The PDS data set (...PREP.JCLCNTL) contains JCL to help in the setup and execution of these utilities.

If you plan to use the DB2 quick start utility, you must BIND the utility first. Use the JCL member BLXDBQ#1 in the PDS data set (...PREP.JCLCNTL) to perform the BIND as a batch job. The DB2 quick start utility has already been preprocessed and prepared. The DBRM is delivered in the Samples data set (...BLSAMP). Once the BIND is completed, the utility is ready for use.

The COBOL, VISION:Inquiry, and VISION:Results quick start utilities do not require any setup, but you can link edit interfaces with the utility for access to source code management libraries such as CA-Panvalet and CA-Librarian. Sample JCL (BLXCBQ#1, BLXCBQ#2, BLXRSQ#1, and BLXRSQ#2) is provided in the PDS data set (...PREP.JCLCNTL) for the optional link edits.

The JCL members for running the utilities are: BLXCBQ#3, BLXDBQ#2, BLXINQ#1, and BLXRSQ#3.

# Maintenance and Support

---

During the life of a VISION:Builder Release, PTFs, formerly known as System Modifications (SMs), are developed to enhance, maintain, and customize the product and components. Any problems that arise are fixed by PTFs (SMs), which are numbered in sequence as they are developed for each release, beginning with 200. There are other patches called APARs, formerly known as Restricted System Modifications (RSMs), that are special customizations to the product and do not apply to all sites. The user should always review the APAR description carefully before applying them to a system.

The PTFs and APARs are identified by component and number using the following format:

CCNNNNN

where:

CC is the Component Identifier:

BL	VISION:Builder engine
CL	COMLIB component
WB	Workbench for ISPF component

NNNNN is the Modification Number Identifier:

00001 to 00199	Numbers assigned to APARs, special patches
00200 to 00500	Numbers assigned to PTFs, general patches

Examples: BL00200, BL00125, CL00215, WB00201.

## Maintenance – Installing the PTFs and APARs

All PTFs and APARs are installed to VISION:Builder and its components under the control of SMP/E. The SMP/E process for handling PTFs and APARs has the following basic steps:

1. Record and save the PTF or APAR into the global zone using the RECEIVE command.
2. Use the APPLY command to install the PTF or APAR to the target libraries.
3. Use the ACCEPT command to install the PTF or APAR into the distribution libraries.

The PTFs are general modifications that are designed for all users and all systems. These should always be installed into VISION:Builder and its components in order to keep the system up to date. PTFs should be installed in both the Target and Distribution Libraries.

The APARs are special modifications that are designed for unique situations. The APARs do not apply to all users and systems. The control statements in the PDS data set (...PREP.JCLCNTL) contain comments for each item that describe the situation addressed by the PTF or APAR. Review the description of any APAR you are considering for your system. Contact Computer Associates Technical Support if you have any questions, concerns, or if you just need more information regarding an APAR. See [Contacting Computer Associates on page 1-11](#).

When installing APARs, there may be some time between the APPLY to the target libraries and the ACCEPT to the distribution libraries. You should take this time to evaluate whether the APAR satisfies the special need for your system. If you decide that the APAR is not appropriate, you can use an SMP/E RESTORE command to remove the APAR from the target libraries. Additionally, you can use an SMP/E REJECT command to remove the APAR for the global zone.

**Note:** Once you ACCEPT an element, such as APAR or PTF, into the distribution libraries, there is no direct method for restoring the previous version of an element in your target libraries.

The PDS data set (...PREP.JCLCNTL) contains some model jobs for performing the various maintenance activities described above. Here are the member names and their functions:

Member Names	Description
BLSMPE#D	RECEIVE a PTF or APAR into the Global Zone and Libraries
BLSMPE#E	APPLY a PTF or APAR into the Target Libraries

Member Names	Description
BLSMPE#F	ACCEPT a PTF or APAR into the Distribution Libraries
BLSMPE#G	RESTORE (remove) a PTF or APAR from the Target Libraries
BLSMPE#H	REJECT (remove) a PTF or APAR from the Global Zone and Libraries

There are other Tools and Facilities that are available for invoking SMP/E commands and functions. Any of these will work because VISION:Builder only uses the standard SMP/E processes. The Systems Group at each site has their favorite tools and procedures, and any of those should work just fine.

**Note:** The APAR runs get a return code of 4 from APPLY and ACCEPT runs because they do not contain prerequisites for other PTFs and APARs.

When PTF runs are performed after an APAR has been processed, they get a return code of 4 because the PTF will not contain prerequisites for any APARs. Remember, the APARs can be identified by their number, which is in the range of cc00001 to cc00199, with the cc being a component identifier.

## Support – Problem Reporting

When a problem is encountered, contact Computer Associates Technical Support to have a representative review your situation. You must provide details to the representative concerning what actions were being performed at the time the problem occurred. Any information on recreating the problem is very useful. Capture any messages or information displayed and communicate these messages to the support representative.

See [Contacting Computer Associates on page 1-11](#).

### VISION:Builder and COMLIB Problem Reporting

For VISION:Builder and COMLIB problems, a Diagnostic Information Page may appear as part of the termination handling and message MK4S701 starts the display. Save the information on this display to send to your Computer Associates Technical Support Representative. In some cases, a complete SYSUDUMP taken at the time the problem occurred may be needed to help determine the reason for the error.

## VISION:Workbench for DOS Problem Reporting

For VISION:Workbench for DOS problems, use the screen print feature to capture any messages displayed on the screen. In some cases, a copy on diskette of the application and/or the definitions being used at the time of the error may be needed to help determine the reason for the error.

## VISION:Workbench for ISPF Problem Reporting

For VISION:Workbench for ISPF problems, gather as much of the following information as possible:

- The objective of your session.
- The name, identification, or description of the last valid panel you saw before the problem.
- Any error messages that were displayed.
- Information from the unexpected error display, if applicable.
- Any other information you feel would be useful in recreating the situation.

### Panel Identification

The upper-left corner of every VISION:Workbench for ISPF data entry panel contains a panel identification name. This name is used to reference the panels in VISION:Workbench for ISPF documentation. This identification name is not the name of the panel member in your panel library.

To view the actual member name of a panel, type `PANELID` on the ISPF command line. To turn this feature off and return to the panel identification name, type `PANELID`. The `PANELID` command toggles between on and off.

### Unexpected Error Panel

If VISION:Workbench for ISPF abnormally terminates or detects a serious internal error, it displays an Unexpected Error panel. This panel contains information that is always useful when trying to track down the cause of the problem.

If you encounter this screen, obtain a screen print of the display before continuing. If you cannot obtain a screen print, record the following information:

- Error messages on the display.
- PSW value.
- The contents of registers 12, 14, and 15.

Use the Help Primary Command to display any message that might be pending.

## User Code

The extensive error checking and data validation techniques built into the VISION:Builder system and components ensure processing integrity. However, these systems have no capacity for determining the integrity of user code incorporated into the application through the facilities of GDBI, GSI, or own-code. Consequently, an error in user code could cause system failure.

Determining the true cause of errors within user code is not easy, can be time consuming, and can significantly increase the cost of maintaining these systems. As a courtesy, Computer Associates works with the customer to help discover where the problem might exist in the user code. Once the nature of the problem in the user code has been determined, it is up to the customer to make the corrections.



This appendix contains an alphabetical list of all the JCL members referenced in the installation procedures, with the complete detail of each member.

<a href="#"><u>BLCOPY1</u></a>	This member copies system tape file number 1 to a disk data set.
<a href="#"><u>BLCOPY2</u></a>	This member copies system tape files 2-16 to disk data sets.
<a href="#"><u>BLSMPE#A</u></a>	This member applies SYSMODs (APARs) into the target zone and target libraries using an in-stream procedure executed once per APAR.
<a href="#"><u>BLSMPE#B</u></a>	This member restores (removes) APARs from the target zone and target libraries using an in-stream procedure executed once per APAR.
<a href="#"><u>BLSMPE#C</u></a>	This member accepts APARs into the distribution zone and distribution libraries using an in-stream procedure executed once per APAR.
<a href="#"><u>BLSMPE#D</u></a>	This member is used as a model to receive PTF/APAR SYSMODs into the global zone and global data sets.
<a href="#"><u>BLSMPE#E</u></a>	This member is used as a model to apply PTF/APAR SYSMODs into the target zone and target libraries using an in-stream procedure executed once per item.
<a href="#"><u>BLSMPE#F</u></a>	This member is used as a model to accept PTF/APAR SYSMODs into the distribution zone and distribution libraries using an in-stream procedure executed once per item.
<a href="#"><u>BLSMPE#G</u></a>	This member is used as a model to restore (remove) PTF/APAR SYSMODs from the target zone and target libraries using an in-stream procedure executed once per item.
<a href="#"><u>BLSMPE#H</u></a>	This member is used as a model to reject (remove) PTF/APAR SYSMODs from the global zone and global data sets using an in-stream procedure executed once per item.
<a href="#"><u>BLSMPE#I</u></a>	This member allocates all the SMP/E and product data sets needed to install and maintain VISION:Builder.

---

<a href="#"><u>BLSMPE#2</u></a>	This member defines the SMP/E CSI and the global, distribution, and target zones.
<a href="#"><u>BLSMPE#3</u></a>	This member receives the modification control statements (MCS) and the elements (SYSMODs) into the global zone and global data sets.
<a href="#"><u>BLSMPE#4</u></a>	This member receives the PTF and APAR SYSMODs into the global zone and global data sets.
<a href="#"><u>BLSMPE#5</u></a>	This member applies the elements (modules) into the target zone and target libraries.
<a href="#"><u>BLSMPE#6</u></a>	This member applies the PTFs into the target zone and target libraries using an in-stream procedure executed once per PTF.
<a href="#"><u>BLSMPE#7</u></a>	This member verifies the installation process run. This job stream is used to demonstrate to the installer that the standard VISION:Builder installation was successful. Several different job steps are run to perform a variety of functions indicating that the standard product is operational.
<a href="#"><u>BLSMPE#8</u></a>	This member accepts the elements (modules) into the distribution zone and distribution libraries.
<a href="#"><u>BLSMPE#9</u></a>	This member accepts the PTFs into the distribution zone and distribution libraries using an in-stream procedure executed once per PTF.
<a href="#"><u>BLXASM#1</u></a>	This member assembles and links the parameter modules M4PARAMS, M4SFARM, M4LEPARM, and MARKLIBP.
<a href="#"><u>BLXASM#2</u></a>	This member assembles and links the parameter modules OQLPARM and BQLPARM.
<a href="#"><u>BLXBAN#1</u></a>	This member displays a signon banner page.
<a href="#"><u>BLXCBO#1</u></a>	This member links the COBOL Quick Start utility with the CA-Librarian interface modules.
<a href="#"><u>BLXCBO#2</u></a>	This member links the COBOL Quick Start utility with the CA-Panvalet interface modules.
<a href="#"><u>BLXCBO#3</u></a>	This member runs the COBOL Quick Start utility.
<a href="#"><u>BLXCOP#1</u></a>	This member copies the target load library to a new user load library.
<a href="#"><u>BLXCOP#2</u></a>	This member copies the VISION:Builder Target System Load Library to an existing user load library.
<a href="#"><u>BLXDBQ#1</u></a>	This member binds the DB2 Quick Start utility.
<a href="#"><u>BLXDBQ#2</u></a>	This member executes the DB2 Quick Start utility.

---

<a href="#"><u>BLXDB2#T</u></a>	This member prepares the MARKSQL module for Teradata table access.
<a href="#"><u>BLXDB2#1</u></a>	This member prepares the MARKSQL module for the various DB2 attach facilities.
<a href="#"><u>BLXDB2#2</u></a>	This member binds the prepared MARKSQL modules used for attaching to DB2 during processing runs.
<a href="#"><u>BLXINQ#1</u></a>	This member runs the VISION:Inquiry <sup>®</sup> Quick Start utility.
<a href="#"><u>BLXMSG#1</u></a>	This member copies the VISION:Builder message modules for use in loading the system LPA.
<a href="#"><u>BLXOLX#1</u></a>	This member copies the online executive help members to a TSO help data set.
<a href="#"><u>BLXOLX#2</u></a>	This member copies some of the online executive command processing modules to the SYS1.LINKLIB.
<a href="#"><u>BLXPAL#1</u></a>	This member catalogs PAL (program analyzer) definitions and processing requests into a M4LIB.
<a href="#"><u>BLXRLK#1</u></a>	This member relinks the VISION:Builder overlay module MARKIV with a user M4OWN module for static own code integration.
<a href="#"><u>BLXRSQ#1</u></a>	This member links the VISION:Results <sup>™</sup> Quick Start utility with the CA-Librarian interface modules.
<a href="#"><u>BLXRSQ#2</u></a>	This member links the results Quick Start utility with CA-Panvalet interface modules.
<a href="#"><u>BLXRSQ#3</u></a>	This member runs the VISION:Results Quick Start utility.

## BLCOPY1

```

// * MEMBER BLCOPY1
// *****
// *
// * THIS JOB COPIES FILE 1
// * FROM THE VISION:UILDER SYSTEM TAPE TO DISK
// *
// * BEFORE YOU RUN THIS JOB, REVIEW JCL AND SPECIFY:
// *
// * THE INPUT TAPE INFORMATION: UNIT, VOLUME SERIAL NUMBER
// * THE OUTPUT DISK DATASET NAME, UNIT AND VOLUME SERIAL NUMBER.
// *
// *****
// *
// *
// COPY1 EXEC PGM=IEBCOPY,REGION=2M
// SYSPRINT DD SYSOUT=*
// *
// INPUT DD DSN=VISION.BUILDER.FILE1,
// DISP=OLD,
// LABEL=(1,SL,EXPDT=98000),
// UNIT=CART,
// VOL=(PRIVATE,RETAIN,SER=(TAPVOL))

```

---

## BLCOPY1 (cont.)

```
//*
//OUTPUT DD DSN=BUILDER.R140.WORK.PDS,
//        DISP=(NEW,CATLG),
//        UNIT=SYSDA,
//        VOL=SER=DSKVOL,
//        SPACE=(TRK,(3,1,1)),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//*
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,15)
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,15)
//SYSIN DD *
//      COPY INDD=INPUT,OUTDD=OUTPUT
//*
```

## BLCOPY2

```
//* MEMBER BLCOPY2
//*****
//*
//* TRANSFER THE VISION:BUILDER SYSTEM TAPE FILES (NUMBER 2 - 16)
//* FROM THE SYSTEM TAPE TO INDIVIDUAL DISK DATA SETS.
//*
//*****
//*
//* THIS PROCEDURE IS REFERENCED IN THE SUBSEQUENT COPY JOB STEPS
//*
//*
//COPY PROC DSNHLQ='BUILDER.R140', HIGH-LEVEL QUALIFIER
//        DSNNAME=, DATA SET NAME (DO NOT CHANGE)
//        DUNIT=SYSDA, DISK UNIT
//        DVOLSER=DISKVOL, DISK VOLSER
//        DTRKS=, DATA SET TRACKS (MINIMUM SHOWN)
//        DDCB=, DATA SET DCB
//        TUNIT=CART, TAPE UNIT
//        TVOLSER=VVVVVV, TAPE VOLSER
//        TFILENO= TAPE FILE NUMBER
//*
//COPY EXEC PGM=IEBCOPY,REGION=2M
//SYSPRINT DD SYSOUT=*
//IN DD DSN=VISION.BUILDER.FILE&TFILENO,DISP=OLD,
//        UNIT=&TUNIT,LABEL=(&TFILENO,SL,EXPDT=98000),
//        VOL=(PRIVATE,RETAIN,SER=(&TVOLSER))
//OUT DD DSN=&DSNHLQ..&DSNAME,
//        DISP=(NEW,CATLG),
//        UNIT=&DUNIT,
//        VOL=SER=&DVOLSER,
//        SPACE=(TRK,&DTRKS),
//        DCB=&DDCB
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,1)
// PEND
//*
//*****
//*
//* THE FOLLOWING JOB STEPS TRANSFER THE
//* VISION:BUILDER SYSTEM TAPE FILES TO DISK DATA SETS.
//*
//* >>>>> NOTE - THE DISK DATASETS ARE ALLOCATED HERE <<<<<
//* >>>>> WITH "DISP=(NEW,CATLG)". <<<<<
//*
//* EACH STEP INVOKES THE INSTREAM PROCEDURE
//* WHICH USES THE FOLLOWING SYMBOLICS:
//*
//* DSNHLQ - HIGH-LEVEL QUALIFIER FOR ALL DATA SETS.
//* (SOME OF THE DATA SETS FROM THE SYSTEM TAPE
//* WILL BE SMP/E INDIRECT DATA SETS.)
//*
//* DSNNAME - DISK DATA SET NAME (DO NOT CHANGE)
//*
```

---

## BLCOPY2 (cont.)

```
//* DUNIT - UNIT TYPE FOR THE DASD. THE DEFAULT IS SYSDA.
//* DVOLSER - VOLUME SERIAL NUMBER OF THE DASD UNIT.
//* (IF NOT NEEDED, REMOVE FROM PROCEDURE AND JOB STEPS.)
//* DTRKS - SPACE ALLOCATION FOR THE DASD DATA SET.
//* (MINIMUM REQUIREMENTS FOR 3390 DEVICES SHOWN.)
//* DDCB - DASD DATA SET DCB VALUES.
//*
//* TUNIT - UNIT TYPE FOR THE TAPE DEVICE. THE DEFAULT IS CART.
//* TVOLSER - VOLUME SERIAL NUMBER OF THE SYSTEM INSTALLATION TAPE.
//* SEE THE EXTERNAL LABEL OF TAPE FOR THE SERIAL NUMBER.
//* TFILENO - THE TAPE FILE NUMBER BEING TRANSFERRED.
//*
//*****
//* FILE2 - COPY THE PREPARATION "CLIST" LIBRARY TO DISK
//*****
//FILE2 EXEC COPY,
//      DSNAME='PREP.CLIST',
//      DUNIT=SYSDA,
//      DVOLSER=DISKVOL,
//      DTRKS='(5,1,5)',
//      DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//      TFILENO=2
//SYSIN DD *
//      COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE3 - COPY THE PREPARATION "PANELS" LIBRARY TO DISK
//*****
//FILE3 EXEC COPY,
//      DSNAME='PREP.PANELS',
//      DUNIT=SYSDA,
//      DVOLSER=DISKVOL,
//      DTRKS='(5,1,10)',
//      DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//      TFILENO=3
//SYSIN DD *
//      COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE4 - COPY THE PREPARATION "MSGs" LIBRARY TO DISK
//*****
//FILE4 EXEC COPY,
//      DSNAME='PREP.MSGS',
//      DUNIT=SYSDA,
//      DVOLSER=DISKVOL,
//      DTRKS='(5,1,5)',
//      DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//      TFILENO=4
//SYSIN DD *
//      COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE5 - COPY THE PREPARATION "SKELS" LIBRARY TO DISK
//*****
//FILE5 EXEC COPY,
//      DSNAME='PREP.SKELS',
//      DUNIT=SYSDA,
//      DVOLSER=DISKVOL,
//      DTRKS='(15,1,15)',
//      DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//      TFILENO=5
//SYSIN DD *
//      COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE6 - COPY THE INSTALL JCL AND CONTROL STATEMENT LIBRARY TO DISK
//*****
//FILE6 EXEC COPY,
//      DSNAME='INSTALL.JCLCNTL',
//      DUNIT=SYSDA,
//      DVOLSER=DISKVOL,
```

---

## BLCOPY2 (cont.)

```
//          DTRKS=' (20,5,20) ',
//          DDCB=' (RECFM=FB,LRECL=80,BLKSIZE=0) ',
//          TFILENO=6
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE7 - COPY THE BUILDER LOAD LIBRARY TO DISK
//*          (SMP/E INDIRECT LIBRARY)
//*****
//FILE7      EXEC COPY,
//          DSNAME='SMPE.I.BLOAD',
//          DUNIT=SYSDA,
//          DVOLSER=DISKVOL,
//          DTRKS=' (90,15,50) ',
//          DDCB=' (RECFM=U,LRECL=0,BLKSIZE=32760) ',
//          TFILENO=7
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE8 - COPY THE BUILDER SAMPLE LIBRARY TO DISK
//*          (SMP/E INDIRECT LIBRARY)
//*****
//FILE8      EXEC COPY,
//          DSNAME='SMPE.I.BLSAMP',
//          DUNIT=SYSDA,
//          DVOLSER=DISKVOL,
//          DTRKS=' (50,5,10) ',
//          DDCB=' (RECFM=FB,LRECL=80,BLKSIZE=0) ',
//          TFILENO=8
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE9 - COPY THE BUILDER COMLIB LOAD LIBRARY TO DISK
//*          (SMP/E INDIRECT LIBRARY)
//*****
//FILE9      EXEC COPY,
//          DSNAME='SMPE.I.CLLOAD',
//          DUNIT=SYSDA,
//          DVOLSER=DISKVOL,
//          DTRKS=' (50,5,15) ',
//          DDCB=' (RECFM=U,LRECL=0,BLKSIZE=32760) ',
//          TFILENO=9
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE10 - COPY THE BUILDER ISPF WORKBENCH LOAD LIBRARY TO DISK
//*          (SMP/E INDIRECT LIBRARY)
//*****
//FILE10     EXEC COPY,
//          DSNAME='SMPE.I.WBLOAD',
//          DUNIT=SYSDA,
//          DVOLSER=DISKVOL,
//          DTRKS=' (50,5,40) ',
//          DDCB=' (RECFM=U,LRECL=0,BLKSIZE=32760) ',
//          TFILENO=10
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//*
//*****
//* FILE11 - COPY THE BUILDER ISPF WORKBENCH CLIST LIBRARY TO DISK
//*          (SMP/E INDIRECT LIBRARY)
//*****
//FILE11     EXEC COPY,
//          DSNAME='SMPE.I.WBCLIST',
//          DUNIT=SYSDA,
//          DVOLSER=DISKVOL,
//          DTRKS=' (20,5,10) ',
```

## BLCOPY2 (cont.)

```
//          DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//          TFILENO=11
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//          *
//          *****
//          * FILE12 - COPY THE BUILDER ISPF WORKBENCH PANEL LIBRARY TO DISK
//          *          (SMP/E INDIRECT LIBRARY)
//          *          *****
//          FILE12 EXEC COPY,
//                  DSNNAME='SMPE.I.WBPANEL',
//                  DUNIT=SYSDA,
//                  DVOLSER=DISKVOL,
//                  DTRKS='(120,5,250)',
//                  DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//                  TFILENO=12
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//          *
//          *****
//          * FILE13 - COPY THE BUILDER ISPF WORKBENCH MSGS LIBRARY TO DISK
//          *          (SMP/E INDIRECT LIBRARY)
//          *          *****
//          FILE13 EXEC COPY,
//                  DSNNAME='SMPE.I.WBMSG',
//                  DUNIT=SYSDA,
//                  DVOLSER=DISKVOL,
//                  DTRKS='(15,1,40)',
//                  DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//                  TFILENO=13
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//          *
//          *****
//          * FILE14 - COPY THE BUILDER ISPF WORKBENCH SKELS LIBRARY TO DISK
//          *          (SMP/E INDIRECT LIBRARY)
//          *          *****
//          FILE14 EXEC COPY,
//                  DSNNAME='SMPE.I.WBSKELS',
//                  DUNIT=SYSDA,
//                  DVOLSER=DISKVOL,
//                  DTRKS='(5,1,5)',
//                  DDCB='(RECFM=FB,LRECL=80,BLKSIZE=0)',
//                  TFILENO=14
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//          *
//          *****
//          * FILE15 - COPY THE BUILDER USER SAS/C RUNTIME LIBRARY TO DISK
//          *          *****
//          FILE15 EXEC COPY,
//                  DSNNAME='SMPE.I.SCLINK',
//                  DUNIT=SYSDA,
//                  DVOLSER=DISKVOL,
//                  DTRKS='(90,1,30)',
//                  DDCB='(RECFM=U,LRECL=0,BLKSIZE=32760)',
//                  TFILENO=15
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//          *
//          *****
//          * FILE16 - COPY THE BUILDER USER SAMPLES LIBRARY TO DISK
//          *          *****
//          FILE16 EXEC COPY,
//                  DSNNAME='USER.EXAMPLES',
//                  DUNIT=SYSDA,
//                  DVOLSER=DISKVOL,
```

---

## BLCOPY2 (cont.)

```
//          DTRKS=' (20,5,10) ',
//          DDCB=' (RECFM=FB,LRECL=80,BLKSIZE=0) ',
//          TFILENO=16
//SYSIN      DD *
//          COPY INDD=IN,OUTDD=OUT
//*
```

## BLSMPE#A

```
//BLSMPE#A JOB (ACCT)
//*
//*  DEFAULT JCL
//*
//*  BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//*  BLSMPE#A - APPLY SYSMODS (APARS) INTO THE TARGET ZONE/LIBRARIES
//*            - USING AN IN-STREAM PROCEDURE EXECUTED ONCE PER APAR
//*****
//*
//*  APPLY SYSMODS (APARS) INSTREAM PROCEDURE
//*
//APPLY      PROC
//*
//*  APPLY AN APAR - EXPECTED RETURN CODE: 0004
//*
//APPLY      EXEC PGM=GIMSMP,REGION=4M
//SMPCSI     DD DSN=BUILDER.R140.CSI,
//            DISP=SHR
//SMPSDDS    DD DSN=BUILDER.R140.SMPSDDS,
//            DISP=SHR
//SMPSSTS    DD DSN=BUILDER.R140.SMPSSTS,
//            DISP=SHR
//SMPMTS     DD DSN=BUILDER.R140.SMPMTS,
//            DISP=SHR
//SMPPTS     DD DSN=BUILDER.R140.SMPPTS,
//            DISP=SHR
//
//          PEND
//*
//*  MODEL FOR INVOKING THE INSTREAM PROCEDURE
//*
//*  CHANGE THE "BLNNNNN" TO THE APAR ID YOU ARE APPLYING
//*  ONLY APPLY ONE (1) APAR PER JOB STEP
//*
//*  RETURN CODE 0004 IS EXPECTED
//*  NO "PREREQUISITES" ARE SPECIFIED
//*
//BLNNNNN EXEC APPLY
//SMPCNTL    DD *
//          SET BDY(BL140TZ) .
//          APPLY
//          SELECT (BLNNNNN) .
//*
//
```

## BLSMPE#B

```
//BLSMPE#B JOB (ACCT)
//*
//*  DEFAULT JCL
//*
//*  BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//*  BLSMPE#B - RESTORE (REMOVE) APARS FROM THE TARGET ZONE/LIBRARIES
```

---

### BLSMPE#B (cont.)

```
//*          - USING AN IN-STREAM PROCEDURE EXECUTED ONCE PER APAR
//*****
//*
//* REMOVE (RESTORE) SYSMODS (APARS) INSTREAM PROCEDURE
//*
//RESTORE PROC
//*
//* REMOVE AN APAR - EXPECTED RETURN CODE: 0000
//*
//RESTORE EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
//        DISP=SHR
//SMPSCDS DD DSN=BUILDER.R140.SMPSCDS,
//        DISP=SHR
//SMPSTS DD DSN=BUILDER.R140.SMPSTS,
//        DISP=SHR
//SMPMTS DD DSN=BUILDER.R140.SMPMTS,
//        DISP=SHR
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,
//        DISP=SHR
//
//        PEND
//*
//* MODEL FOR INVOKING THE INSTREAM PROCEDURE
//*
//* CHANGE "BLNNNNN" TO THE APAR ID YOU ARE REMOVING
//* ONLY REMOVE ONE (1) APAR PER JOB STEP
//*
//BLNNNNN EXEC RESTORE
//SMPCNTL DD *
//        SET BDY(BL140TZ).
//        RESTORE
//        SELECT (BLNNNNN) .
//
//
//
```

### BLSMPE#C

```
//BLSMPE#C JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLSMPE#C - ACCEPT APARS INTO THE DISTRIBUTION ZONE/LIBRARIES
//*          - USING AN IN-STREAM PROCEDURE EXECUTED ONCE PER APAR
//*****
//*
//* ACCEPT SYSMODS (APARS) INSTREAM PROCEDURE
//*
//ACCEPT PROC
//*
//* ACCEPT AN APAR - EXPECTED RETURN CODE: 0000
//*
//ACCEPT EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
//        DISP=SHR
//SMPSCDS DD DSN=BUILDER.R140.SMPSCDS,
//        DISP=SHR
//SMPSTS DD DSN=BUILDER.R140.SMPSTS,
//        DISP=SHR
//SMPMTS DD DSN=BUILDER.R140.SMPMTS,
//        DISP=SHR
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,
//        DISP=SHR
//
//        PEND
//*
//* MODEL FOR INVOKING THE INSTREAM PROCEDURE
```

---

## BLSMPE#C (cont.)

```
//*
//*  CHANGE THE "BLNNNNN" TO THE APAR ID YOU ARE ACCEPTING
//*  ONLY ACCEPT ONE (1) APAR PER JOB STEP
//*
//BLNNNNN EXEC ACCEPT
//SMPCNTL  DD *
//      SET BDY(BL140DZ) .
//      ACCEPT
//          SELECT (BLNNNNN) .
//
//*
```

## BLSMPE#D

```
//BLSMPE#D JOB (ACCT)
//*
//*  DEFAULT JCL
//*
//*  BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//*  BLSMPE#D - A MODEL TO RECEIVE PTF/APAR SYSMODS INTO THE
//*              - GLOBAL ZONE/DATA SETS
//*****
//*
//*  THIS IS A MODEL JOB FOR
//*  RECEIVING PTFS AND/OR APARS
//*  INTO THE GLOBAL CSI
//*
//*  AT THE "SMPPTFIN" DD, POINT TO THE LOCATION OF THE PTF AND/OR APAR
//*  SMP/E MCS CONTROL STATEMENTS.
//*
//RECEIVE EXEC PGM=GIMSMP,REGION=4M
//SMPCSI  DD DSN=BUILDER.R140.CSI,
//          DISP=SHR
//SMPSCDS DD DSN=BUILDER.R140.SMPSCDS,
//          DISP=SHR
//SMPSTS  DD DSN=BUILDER.R140.SMPSTS,
//          DISP=SHR
//SMPMTS  DD DSN=BUILDER.R140.SMPMTS,
//          DISP=SHR
//SMPPTS  DD DSN=BUILDER.R140.SMPPTS,
//          DISP=SHR
//SMPLOG  DD DSN=BUILDER.R140.SMPLOG,
//          DISP=SHR
//SMPLOGA DD DSN=BUILDER.R140.SMPLOGA,
//          DISP=SHR
//SMPPTFIN DD DSN=THE.PTFS.AND.OR.APAR.STATEMENTS,
//          DISP=SHR
//SMPCNTL DD *
//          SET BDY(GLOBAL) .
//          RECEIVE SYSMODS LIST.
//          LIST.
//
//*
```

## BLSMPE#E

```
//BLSMPE#E JOB (ACCT)
//*
//*  DEFAULT JCL
//*
//*  BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
```

---

### BLSMPE#E (cont.)

```
/* BLSMPE#E - A MODEL TO APPLY PTF/APAR SYSMODS INTO THE
/*          - TARGET ZONE/LIBRARIES USING AN IN-STREAM
/*          - PROCEDURE EXECUTED ONCE PER ITEM
/******
/*
/* THIS IS A MODEL JOB FOR
/* APPLYING PTFS AND/OR APARS
/* INTO THE TARGET LIBRARY
/*
/* APPLY SYSMODS (PTFS/APARS) INSTREAM PROCEDURE
/*
/* APPLY   PROC
/*
/*APPLSTP EXEC PGM=GIMSMP,REGION=4M
/*SMPCSI   DD DSN=BUILDER.R140.CSI,
/*          DISP=SHR
/*SMPCSDS  DD DSN=BUILDER.R140.SMPSCDS,
/*          DISP=SHR
/*SMPSTS   DD DSN=BUILDER.R140.SMPSTS,
/*          DISP=SHR
/*SMPMTS   DD DSN=BUILDER.R140.SMPMTS,
/*          DISP=SHR
/*SMPPTS   DD DSN=BUILDER.R140.SMPPTS,
/*          DISP=SHR
/*          PEND
/*
/* EXECUTE THE APPLY PROCEDURE ONCE PER PTF AND/OR APAR
/*
/* CHANGE "BLNNNNN" TO THE SYSMOD ID
/* USE THE "CHECK" STATEMENT FOR A "TEST RUN"
/*
/*BLNNNNN EXEC APPLY
/*SMPCNTL  DD *
/*          SET BDY(BL140TZ).
/*          APPLY
/*              CHECK
/*              SELECT (BLNNNNN) .
/*
/*
```

### BLSMPE#F

```
/*BLSMPE#F JOB (ACCT)
/*
/* DEFAULT JCL
/*
/* BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/*
/******
/* BLSMPE#F - A MODEL TO ACCEPT PTF/APAR SYSMODS INTO THE
/*          - DISTRIBUTION ZONE/LIBRARIES USING AN IN-STREAM
/*          - PROCEDURE EXECUTED ONCE PER ITEM
/******
/*
/* THIS IS A MODEL JOB FOR
/* ACCEPTING PTFS AND/OR APARS
/* INTO THE TARGET LIBRARY
/*
/* ACCEPT SYSMODS (PTFS/APARS) INSTREAM PROCEDURE
/*
/* ACCEPT   PROC
/*
/* ACCEPT A PTF - EXPECTED RETURN CODE: 0000
/*
/*ACCEPT EXEC PGM=GIMSMP,REGION=4M
/*SMPCSI   DD DSN=BUILDER.R140.CSI,
/*          DISP=SHR
/*SMPCSDS  DD DSN=BUILDER.R140.SMPSCDS,
```

---

## BLSMPE#F (cont.)

```
//          DISP=SHR
//SMPSTS   DD DSN=BUILDER.R140.SMPSTS,
//          DISP=SHR
//SMPMTS   DD DSN=BUILDER.R140.SMPMTS,
//          DISP=SHR
//SMPPTS   DD DSN=BUILDER.R140.SMPPTS,
//          DISP=SHR
//          PEND
//*
//* EXECUTE THE ACCEPT PROCEDURE ONCE PER PTF AND/OR APAR
//*
//* CHANGE "BLNNNNN" TO THE SYSMOD ID
//*
//BLNNNNN EXEC ACCEPT
//SMPCNTL  DD *
//          SET BDY(BL140DZ) .
//          ACCEPT
//          SELECT (BLNNNNN) .
//*
//
```

## BLSMPE#G

```
//BLSMPE#G JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLSMPE#G - A MODEL TO RESTORE (REMOVE) PTF/APAR SYSMODS FROM
//*             - THE TARGET ZONE/LIBRARIES USING AN IN-STREAM
//*             - PROCEDURE EXECUTED ONCE PER ITEM
//*****
//*
//* RESTORE (REMOVE) SYSMODS INSTREAM PROCEDURE
//*
//RESTORE PROC
//*
//RESTORE EXEC PGM=GIMSMP,REGION=4M
//SMPCSI   DD DSN=BUILDER.R140.CSI,
//          DISP=SHR
//SMPSDDS  DD DSN=BUILDER.R140.SMPSDDS,
//          DISP=SHR
//SMPSTS   DD DSN=BUILDER.R140.SMPSTS,
//          DISP=SHR
//SMPMTS   DD DSN=BUILDER.R140.SMPMTS,
//          DISP=SHR
//SMPPTS   DD DSN=BUILDER.R140.SMPPTS,
//          DISP=SHR
//          PEND
//*
//* MODEL FOR INVOKING THE INSTREAM PROCEDURE
//*
//* CHANGE "BLNNNNN" TO THE SYSMOD ID
//* ONLY REMOVE ONE (1) SYSMOD PER JOB STEP
//*
//BLNNNNN EXEC RESTORE
//SMPCNTL  DD *
//          SET BDY(BL140TZ) .
//          RESTORE
//          SELECT (BLNNNNN) .
//*
//
```

---

## BLSMPE#H

```
//BLSMPE#G JOB (ACCT)
//*
//* DEFAULT JCL
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*****
//* BLSMPE#H - A MODEL TO REJECT (REMOVE) PTF/APAR SYSMODS FROM
//*           - THE GLOBAL ZONE/DATA SETS USING AN IN-STREAM
//*           - PROCEDURE EXECUTED ONCE PER ITEM
//*****
//*
//* REJECT (REMOVE) SYSMODS INSTREAM PROCEDURE
//*
//REJECT  PROC
//*
//REJECT  EXEC  PGM=GIMSMP,REGION=4M
//SMPCSI   DD  DSN=BUILDER.R140.CSI,
//          DISP=SHR
//SMPSCDS  DD  DSN=BUILDER.R140.SMPSCDS,
//          DISP=SHR
//SMPSTS   DD  DSN=BUILDER.R140.SMPSTS,
//          DISP=SHR
//SMPMTS   DD  DSN=BUILDER.R140.SMPMTS,
//          DISP=SHR
//SMPPTS   DD  DSN=BUILDER.R140.SMPPTS,
//          DISP=SHR
//          PEND
//*
//* MODEL FOR INVOKING THE INSTREAM PROCEDURE
//*
//*   CHANGE THE "BLNNNNN" TO THE SYSMOD ID YOU ARE REMOVING
//*   ONLY REMOVE ONE (1) SYSMOD PER JOB STEP
//*
//BLNNNNN EXEC REJECT
//SMPCNTL  DD  *
//          SET BDY(GLOBAL) .
//          REJECT
//          SELECT (BLNNNNN) .
//*
//
```

## BLSMPE#1

```
//BLSMPE#1 JOB (ACCT)
//*
//* DEFAULT JCL
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*****
//* BLSMPE#1 - ALLOCATE ALL THE SMP/E AND PRODUCT DATA SETS NEEDED
//*           - TO INSTALL AND MAINTAIN THE VISION:BUILDER SOFTWARE
//*****
//*
//* ALLOCATE THE SMP/E CSI AND SMPPTS DATA SETS
//*
//*   CSI CREATE - A "NEW" CSI WAS REQUESTED
//*
//*   EXPECTED RETURN CODE: 0000
//*
//CREATE  EXEC  PGM=IDCAMS,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSIN   DD  *
//          DELETE (BUILDER.R140.CSI) CLUSTER
```

## BLSMPE#1 (cont.)

```
SET MAXCC=0
DEFINE CLUSTER (
    NAME(BUILDER.R140.CSI)
    FREESPACE(10,5)
    KEYS(24 0)
    RECORDSIZE(24 143)
    SHAREOPTIONS(2 3)
)
DATA(
    NAME(BUILDER.R140.CSI.DATA)
    CONTROLINTERVALSIZE(4096)
    CYLINDERS(60 20)
)
INDEX(
    NAME(BUILDER.R140.CSI.INDEX)
    CYLINDERS(5 5)
    IMBED
)

/*
/**
/** INITIALIZE THE "NEW" CSI
/**
/** EXPECTED RETURN CODE: 0000
/**
/** INIT EXEC PGM=IDCAMS,REGION=4M
/**SYSPRINT DD SYSOUT=*
/**SMPECSI DD DSN=BUILDER.R140.CSI,
/** DISP=OLD
/**ZPOOL DD DSN=SYS1.MACLIB(GIMZPOOL),DISP=SHR
/**SYSIN DD *
REPRO OUTFILE(SMPECSI) INFILE(ZPOOL)
/*
/**
/** SMPPTS ALLOCATE (DELETE ANY PREVIOUS DATA SET)
/**
/** EXPECTED RETURN CODE: 0000
/**
/**DEL1 EXEC PGM=IEFBR14
/**DD1 DD DSN=BUILDER.R140.SMPPTS,
/** DISP=(MOD,DELETE),
/** SPACE=(TRK,(0,0)),
/** UNIT=SYSDA
/**
/**ALLOC1 EXEC PGM=IEFBR14
/**SMPPTS DD DSN=BUILDER.R140.SMPPTS,
/** DISP=(NEW,CATLG,DELETE),
/** DSNTYPE=LIBRARY,
/** UNIT=SYSDA,
/** SPACE=(CYL,(5,5,50)),
/** DCB=(RECFM=FB,LRECL=80)
/**
/**
/*******
/**
/** ALLOCATE SMP/E WORK DATA SETS ASSOCIATED WITH VISION:BUILDER
/** (DELETE ANY PREVIOUS DATA SETS)
/**
/**DEL2 EXEC PGM=IEFBR14
/**DD1 DD DSN=BUILDER.R140.SMPPTS,
/** DISP=(MOD,DELETE),
/** SPACE=(TRK,(0,0)),
/** UNIT=SYSDA
/**DD2 DD DSN=BUILDER.R140.SMPSCDS,
/** DISP=(MOD,DELETE),
/** SPACE=(TRK,(0,0)),
/** UNIT=SYSDA
/**DD3 DD DSN=BUILDER.R140.SMPSTS,
/** DISP=(MOD,DELETE),
/** SPACE=(TRK,(0,0)),
/** UNIT=SYSDA
/**DD4 DD DSN=BUILDER.R140.SMPLOG,
```

---

### BLSMPE#1 (cont.)

```
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD5      DD DSN=BUILDER.R140.SMPLOGA,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//*
//ALLOC2   EXEC PGM=IEFBR14
//SMPMTS   DD DSN=BUILDER.R140.SMPMTS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(2,1,50)),
//          DCB=(RECFM=FB,LRECL=80)
//SMPSCDS  DD DSN=BUILDER.R140.SMPSCDS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(2,1,50)),
//          DCB=(RECFM=FB,LRECL=80)
//SMPSTS   DD DSN=BUILDER.R140.SMPSTS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(2,1,50)),
//          DCB=(RECFM=FB,LRECL=80)
//SMPLOG   DD DSN=BUILDER.R140.SMPLOG,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(5,2)),
//          DCB=(RECFM=VB,LRECL=510,BLKSIZE=27900)
//SMPLOGA  DD DSN=BUILDER.R140.SMPLOGA,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(5,2)),
//          DCB=(RECFM=VB,LRECL=510,BLKSIZE=27900)
//*
//*****
//*
//* ALLOCATE SMP/E DISTRIBUTION LIBRARIES
//* (DELETE ANY PREVIOUS DATA SETS)
//*
//DEL3     EXEC PGM=IEFBR14
//DD1      DD DSN=BUILDER.R140.SMPE.D.BLSYSL,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD2      DD DSN=BUILDER.R140.SMPE.D.BLSAMP,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD3      DD DSN=BUILDER.R140.SMPE.D.WBCLIST,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD4      DD DSN=BUILDER.R140.SMPE.D.WBPANEL,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD5      DD DSN=BUILDER.R140.SMPE.D.WBMSGGS,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD6      DD DSN=BUILDER.R140.SMPE.D.WBSKELS,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//*
//ALLOC3   EXEC PGM=IEFBR14
//BLSYSL   DD DSN=BUILDER.R140.SMPE.D.BLSYSL,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(285,15,120)),
```

---

## BLSMPE#1 (cont.)

```
//          DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)
//BLSAMP   DD DSN=BUILDER.R140.SMPE.D.BLSAMP,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(50,5,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBCLIST  DD DSN=BUILDER.R140.SMPE.D.WBCLIST,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(20,5,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBPNEL   DD DSN=BUILDER.R140.SMPE.D.WBPNEL,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(120,5,250)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBMSGSGS DD DSN=BUILDER.R140.SMPE.D.WBMSGSGS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,1,40)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBSKELS  DD DSN=BUILDER.R140.SMPE.D.WBSKELS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(5,1,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//*
//*****
//*
//* ALLOCATE SMP/E TARGET LIBRARIES
//* (DELETE ANY PREVIOUS DATA SETS)
//*
//DEL4     EXEC PGM=IEFBR14
//DD1      DD DSN=BUILDER.R140.SMPE.T.BLSYSL,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD2      DD DSN=BUILDER.R140.SMPE.T.BLSAMP,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD3      DD DSN=BUILDER.R140.SMPE.T.WBCLIST,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD4      DD DSN=BUILDER.R140.SMPE.T.WBPNEL,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD5      DD DSN=BUILDER.R140.SMPE.T.WBMSGSGS,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//DD6      DD DSN=BUILDER.R140.SMPE.T.WBSKELS,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=SYSDA
//*
//ALLOC4   EXEC PGM=IEFBR14
//BLSYSL   DD DSN=BUILDER.R140.SMPE.T.BLSYSL,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(285,15,120)),
//          DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)
//BLSAMP   DD DSN=BUILDER.R140.SMPE.T.BLSAMP,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(50,5,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBCLIST  DD DSN=BUILDER.R140.SMPE.T.WBCLIST,
//          DISP=(NEW,CATLG,DELETE),
```

---

## BLSMPE#1 (cont.)

```
//          UNIT=SYSDA,
//          SPACE=(TRK,(20,5,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBPNEL  DD DSN=BUILDER.R140.SMPE.T.WBPNEL,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(120,5,250)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBMGS   DD DSN=BUILDER.R140.SMPE.T.WBMGS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,1,40)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//WBSKELS DD DSN=BUILDER.R140.SMPE.T.WBSKELS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(5,1,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
// *
//
```

## BLSMPE#2

```
//BLSMPE#2 JOB (ACCT)
// *
// * DEFAULT JCL
// *
// * BUILT BY THE INSTALLATION PREPARATION DIALOG
// *
// *
// *****
// * BLSMPE#2 - DEFINE IN THE SMP/E CSI, THE GLOBAL,
// * - DISTRIBUTION, AND TARGET ZONES
// *****
// *
// * DEFINE THE CSI GLOBAL, TARGET AND DISTRIBUTION ZONES
// *
// * EXPECTED RETURN CODE: 00
// *
//SMPE EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
//        DISP=SHR
//SMPLOG DD DSN=BUILDER.R140.SMPLOG,
//        DISP=SHR
//SMPLOGA DD DSN=BUILDER.R140.SMPLOGA,
//        DISP=SHR
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,
//        DISP=SHR
//SMPOUT DD SYSOUT=*
//SMPPUNCH DD SYSOUT=*
//SMPRPT DD SYSOUT=*
//SMPLIST DD SYSOUT=*
//SMPSNAP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD DUMMY
//SMPTLIB DD UNIT=SYSDA,
//        SPACE=(CYL,(5,1))
//SMPWRK1 DD UNIT=SYSDA,
//        SPACE=(CYL,(5,5,50))
//SMPWRK2 DD UNIT=SYSDA,
//        SPACE=(CYL,(5,5,50))
//SMPWRK3 DD UNIT=SYSDA,
//        SPACE=(CYL,(5,5,50))
//SMPWRK4 DD UNIT=SYSDA,
//        SPACE=(CYL,(5,5,50))
//SMPWRK5 DD UNIT=SYSDA,
//        SPACE=(CYL,(5,5,50))
//SMPWRK6 DD UNIT=SYSDA,
//        SPACE=(CYL,(5,5,50))
//
```

---

## BLSMPE#2 (cont.)

```
//SYSUT1 DD UNIT=SYSDA,
//      SPACE=(CYL,(5,2))
//SYSUT2 DD UNIT=SYSDA,
//      SPACE=(CYL,(5,2))
//SYSUT3 DD UNIT=SYSDA,
//      SPACE=(CYL,(5,2))
//SYSUT4 DD UNIT=SYSDA,
//      SPACE=(CYL,(5,2))
//*
//SMPCNTL DD *
SET BDY(GLOBAL) .          /* GLOBAL ZONE DEFINES */
UCLIN.
  ADD GLOBALZONE
  SREL (Z038)
  FMID (CCVC140)
  OPTIONS(BL140OP)
  ZONEDESCRIPTION(VISION:BUILDER RELEASE 14.0)
  ZONEINDEX(
    (BL140DZ,BUILDER.R140.CSI,DLIB)
    (BL140TZ,BUILDER.R140.CSI,TARGET)
  ).
  ADD OPTIONS(BL140OP)
  AMS(AMS)
  ASM(ASSEM)
  COMP(COMPRESS)
  COPY(COPY)
  LKED(LINKEDIT)
  NOPURGE
  NOREJECT
  RETRY(RETRY)
  UPDATE(UPDATE)
  ZAP(IMASPPZAP) .          /* THE OPTIONS ARE SMP/E STD DEFAULTS */
  ADD UTILITY(AMS)
  NAME(IDCAMS) .            /* THIS UTILITY IS THE STD SMP/E DEFAULT */
  ADD UTILITY(ASSEM)
  NAME(ASMA90)
  PARM(XREF,NOOBJECT,DECK)
  RC(04) .                  /* THIS UTILITY IS THE STD SMP/E DEFAULT */
  ADD UTILITY(COMPRESS)
  NAME(IEBCOPY) .           /* THIS UTILITY IS THE STD SMP/E DEFAULT */
  ADD UTILITY(COPY)
  NAME(IEBCOPY) .           /* THIS UTILITY IS THE STD SMP/E DEFAULT */
  ADD UTILITY(LINKEDIT)
  NAME(IEWL)
  PARM(LET,LIST,NCAL,XREF)
  RC(08) .                  /* THIS UTILITY IS THE STD SMP/E DEFAULT */
  ADD UTILITY(RETRY)
  NAME(IEBCOPY) .           /* THIS UTILITY IS THE STD SMP/E DEFAULT */
  ADD UTILITY(UPDATE)
  NAME(IEBUPDTE) .          /* THIS UTILITY IS THE STD SMP/E DEFAULT */
  ADD UTILITY(IMASPPZAP)
  NAME(IMASPPZAP)
  PARM(IGNIDRFULL)
  RC(04) .                  /* ADD'L PARM, OTHERS, STD SMP/E DEFAULT */
  ADD DDDEF(SMPLOG)
  DA (BUILDER.R140.SMPLOG)
  MOD.
  ADD DDDEF(SMPLOGA)
  DA (BUILDER.R140.SMPLOGA)
  MOD.
  ADD DDDEF(SMPPTS)
  DA (BUILDER.R140.SMPPTS)
  OLD.
  ADD DDDEF(SMPOUT) SYSOUT(*) .
  ADD DDDEF(SMPRPT) SYSOUT(*) .
  ADD DDDEF(SYSPRINT) SYSOUT(*) .
  ADD DDDEF(SMPTLIB)
  UNIT(SYSDA) .
  ADD DDDEF(SYSUT1) UNIT(SYSDA)
  CYL SPACE(5,2) NEW DELETE.
  ADD DDDEF(SYSUT2) UNIT(SYSDA)
```

---

## BLSMPE#2 (cont.)

```

                                CYL SPACE(5,2) NEW DELETE.
ADD DDDEF(SYSUT3)              UNIT(SYSDA)
                                CYL SPACE(5,2) NEW DELETE.
ADD DDDEF(SYSUT4)              UNIT(SYSDA)
                                CYL SPACE(5,2) NEW DELETE.
ENDUCL.
SET BDY(BL140DZ) .             /* DISTRIBUTION ZONE DEFINES */
UCLIN.
  ADD DLIBZONE(BL140DZ)
    SREL (Z038)
    RELATED(BL140TZ)
    OPTIONS(BL140OP) .
  ADD DDDEF(SMPSCDS)
    DA (BUILDER.R140.SMPSCDS)
    OLD.
  ADD DDDEF(SMPMTS)
    DA (BUILDER.R140.SMPMTS)
    OLD.
  ADD DDDEF(SMPPTS)
    DA (BUILDER.R140.SMPPTS)
    SHR.
  ADD DDDEF(SMPSTS)
    DA (BUILDER.R140.SMPSTS)
    OLD.
  ADD DDDEF(SMPLOG)
    DA (BUILDER.R140.SMPLOG)
    MOD.
  ADD DDDEF(SMPLOGA)
    DA (BUILDER.R140.SMPLOGA)
    MOD.
  ADD DDDEF(BLTLOAD)
    DA (BUILDER.R140.SMPE.T.BLSYSL)
    SHR.
  ADD DDDEF(CLTLOAD)
    DA (BUILDER.R140.SMPE.T.BLSYSL)
    SHR.
  ADD DDDEF(WBTLOAD)
    DA (BUILDER.R140.SMPE.T.BLSYSL)
    SHR.
  ADD DDDEF(BLTSAMP)
    DA (BUILDER.R140.SMPE.T.BLSAMP)
    SHR.
  ADD DDDEF(WBTPANEL)
    DA (BUILDER.R140.SMPE.T.WBPANEL)
    SHR.
  ADD DDDEF(WBTMSGGS)
    DA (BUILDER.R140.SMPE.T.WBMSGGS)
    SHR.
  ADD DDDEF(WBTSKELS)
    DA (BUILDER.R140.SMPE.T.WBSKELS)
    SHR.
  ADD DDDEF(WBTCLIST)
    DA (BUILDER.R140.SMPE.T.WBCLIST)
    SHR.
  ADD DDDEF(SCTLINK)
    DA (BUILDER.R140.SMPE.T.BLSYSL)
    SHR.
  ADD DDDEF(BLDLOAD)
    DA (BUILDER.R140.SMPE.D.BLSYSL)
    SHR.
  ADD DDDEF(BLDSAMP)
    DA (BUILDER.R140.SMPE.D.BLSAMP)
    SHR.
  ADD DDDEF(CLDLOAD)
    DA (BUILDER.R140.SMPE.D.BLSYSL)
    SHR.
  ADD DDDEF(WBDLOAD)
    DA (BUILDER.R140.SMPE.D.BLSYSL)
    SHR.
  ADD DDDEF(WBDPANEL)
    DA (BUILDER.R140.SMPE.D.WBPANEL)
```

---

## BLSMPE#2 (cont.)

```
      SHR.
ADD DDDEF (WBDMGS)
      DA      (BUILDER.R140.SMPE.D.WBMSG)
      SHR.
ADD DDDEF (WBDSKELS)
      DA      (BUILDER.R140.SMPE.D.WBSKELS)
      SHR.
ADD DDDEF (WBDCLIST)
      DA      (BUILDER.R140.SMPE.D.WBCLIST)
      SHR.
ADD DDDEF (SCDLINK)
      DA      (BUILDER.R140.SMPE.D.BLSYSL)
      SHR.
ADD DDDEF (SYSMAC)
      DA      (SYS1.MACLIB)
      SHR.
ADD DDDEF (RESLIB)
      DA      (IMS.RESLIB)
      SHR.
ADD DDDEF (DSNLOAD)
      DA      (DB2.SDSNLOAD)
      SHR.
ADD DDDEF (SYSLIB)  CONCAT (SYSMAC
                          RESLIB
                          DSNLOAD
                          ) .
ADD DDDEF (SMPOUT)   SYSOUT (*) .
ADD DDDEF (SMPPUNCH) SYSOUT (*) .
ADD DDDEF (SMPRPT)   SYSOUT (*) .
ADD DDDEF (SMPLIST)  SYSOUT (*) .
ADD DDDEF (SMPSNAP)  SYSOUT (*) .
ADD DDDEF (SYSPRINT) SYSOUT (*) .
ADD DDDEF (SYSUDUMP) SYSOUT (*) .
ADD DDDEF (SMPTLIB)  UNIT (SYSDA) .
ADD DDDEF (SMPWRK1)  UNIT (SYSDA)
                     CYL SPACE (5,5) DIR (50) NEW DELETE.
ADD DDDEF (SMPWRK2)  UNIT (SYSDA)
                     CYL SPACE (5,5) DIR (50) NEW DELETE.
ADD DDDEF (SMPWRK3)  UNIT (SYSDA)
                     CYL SPACE (5,5) DIR (50) NEW DELETE.
ADD DDDEF (SMPWRK4)  UNIT (SYSDA)
                     CYL SPACE (5,5) DIR (50) NEW DELETE.
ADD DDDEF (SMPWRK5)  UNIT (SYSDA)
                     CYL SPACE (5,5) DIR (50) NEW DELETE.
ADD DDDEF (SMPWRK6)  UNIT (SYSDA)
                     CYL SPACE (5,5) DIR (50) NEW DELETE.
ADD DDDEF (SYSUT1)   UNIT (SYSDA)
                     CYL SPACE (5,2) NEW DELETE.
ADD DDDEF (SYSUT2)   UNIT (SYSDA)
                     CYL SPACE (5,2) NEW DELETE.
ADD DDDEF (SYSUT3)   UNIT (SYSDA)
                     CYL SPACE (5,2) NEW DELETE.
ADD DDDEF (SYSUT4)   UNIT (SYSDA)
                     CYL SPACE (5,2) NEW DELETE.
ENDUCL.
SET BDY (BL140TZ) .          /* TARGET ZONE DEFINES */
UCLIN.
      ADD TARGETZONE (BL140TZ)
      SREL (Z038)
      RELATED (BL140DZ)
      OPTIONS (BL140OP) .
ADD DDDEF (SMPSCDS)
      DA      (BUILDER.R140.SMPSCDS)
      OLD.
ADD DDDEF (SMPMTS)
      DA      (BUILDER.R140.SMPMTS)
      OLD.
ADD DDDEF (SMPPTS)
      DA      (BUILDER.R140.SMPPTS)
      SHR.
```

---

## BLSMPE#2 (cont.)

```
ADD DDDEF (SMPSTS)
DA      (BUILDER.R140.SMPSTS)
OLD.
ADD DDDEF (SMPLOG)
DA      (BUILDER.R140.SMPLOG)
MOD.
ADD DDDEF (SMPLOGA)
DA      (BUILDER.R140.SMPLOGA)
MOD.
ADD DDDEF (BLDLOAD)
DA      (BUILDER.R140.SMPE.D.BLSYSL)
SHR.
ADD DDDEF (BLDSAMP)
DA      (BUILDER.R140.SMPE.D.BLSAMP)
SHR.
ADD DDDEF (CLDLOAD)
DA      (BUILDER.R140.SMPE.D.BLSYSL)
SHR.
ADD DDDEF (WBDLOAD)
DA      (BUILDER.R140.SMPE.D.BLSYSL)
SHR.
ADD DDDEF (WBDPANEL)
DA      (BUILDER.R140.SMPE.D.WBPANEL)
SHR.
ADD DDDEF (WBDMGS)
DA      (BUILDER.R140.SMPE.D.WBMGS)
SHR.
ADD DDDEF (WBDSKELS)
DA      (BUILDER.R140.SMPE.D.WBSKELS)
SHR.
ADD DDDEF (WBDCLIST)
DA      (BUILDER.R140.SMPE.D.WBCLIST)
SHR.
ADD DDDEF (SCDLINK)
DA      (BUILDER.R140.SMPE.D.BLSYSL)
SHR.
ADD DDDEF (BLTLOAD)
DA      (BUILDER.R140.SMPE.T.BLSYSL)
SHR.
ADD DDDEF (CLTLOAD)
DA      (BUILDER.R140.SMPE.T.BLSYSL)
SHR.
ADD DDDEF (WBTLOAD)
DA      (BUILDER.R140.SMPE.T.BLSYSL)
SHR.
ADD DDDEF (BLTSAMP)
DA      (BUILDER.R140.SMPE.T.BLSAMP)
SHR.
ADD DDDEF (WBTPANEL)
DA      (BUILDER.R140.SMPE.T.WBPANEL)
SHR.
ADD DDDEF (WBTMGS)
DA      (BUILDER.R140.SMPE.T.WBMGS)
SHR.
ADD DDDEF (WBTSKELS)
DA      (BUILDER.R140.SMPE.T.WBSKELS)
SHR.
ADD DDDEF (WBTCLIST)
DA      (BUILDER.R140.SMPE.T.WBCLIST)
SHR.
ADD DDDEF (SCTLINK)
DA      (BUILDER.R140.SMPE.T.BLSYSL)
SHR.
ADD DDDEF (SYSMAC)
DA      (SYS1.MACLIB)
SHR.
ADD DDDEF (RESLIB)
DA      (IMS.RESLIB)
SHR.
ADD DDDEF (DSNLOAD)
DA      (DB2.SDSNLOAD)
```

---

## BLSMPE#2 (cont.)

```
      SHR.
      ADD DDDEF (SYSLIB)  CONCAT (SYSMAC
                                RESLIB
                                DSNLOAD
                                ).
      ADD DDDEF (SMPOUT)  SYSOUT (*).
      ADD DDDEF (SMFPUNCH) SYSOUT (*).
      ADD DDDEF (SMPRPT)  SYSOUT (*).
      ADD DDDEF (SMPLIST)  SYSOUT (*).
      ADD DDDEF (SMPSNAP)  SYSOUT (*).
      ADD DDDEF (SYSPRINT) SYSOUT (*).
      ADD DDDEF (SYSUDUMP) SYSOUT (*).
      ADD DDDEF (SMPTLIB)  UNIT (SYSDA).
      ADD DDDEF (SMPWRK1)  UNIT (SYSDA)
                           CYL SPACE (5,5) DIR (50) NEW DELETE.
      ADD DDDEF (SMPWRK2)  UNIT (SYSDA)
                           CYL SPACE (5,5) DIR (50) NEW DELETE.
      ADD DDDEF (SMPWRK3)  UNIT (SYSDA)
                           CYL SPACE (5,5) DIR (50) NEW DELETE.
      ADD DDDEF (SMPWRK4)  UNIT (SYSDA)
                           CYL SPACE (5,5) DIR (50) NEW DELETE.
      ADD DDDEF (SMPWRK5)  UNIT (SYSDA)
                           CYL SPACE (5,5) DIR (50) NEW DELETE.
      ADD DDDEF (SMPWRK6)  UNIT (SYSDA)
                           CYL SPACE (5,5) DIR (50) NEW DELETE.
      ADD DDDEF (SYSUT1)   UNIT (SYSDA)
                           CYL SPACE (5,2) NEW DELETE.
      ADD DDDEF (SYSUT2)   UNIT (SYSDA)
                           CYL SPACE (5,2) NEW DELETE.
      ADD DDDEF (SYSUT3)   UNIT (SYSDA)
                           CYL SPACE (5,2) NEW DELETE.
      ADD DDDEF (SYSUT4)   UNIT (SYSDA)
                           CYL SPACE (5,2) NEW DELETE.
      ENDUCL.
/*
//
```

## BLSMPE#3

```
//BLSMPE#3 JOB (ACCT)
/*
/* * DEFAULT JCL
/* *
/* * BUILT BY THE INSTALLATION PREPARATION DIALOG
/* *
/* *
/* *****
/* * BLSMPE#3 - RECEIVE THE MODIFICATION CONTROL STATEMENTS (MCS) AND
/* *           - THE ELEMENTS (SYSMODS) INTO THE GLOBAL ZONE/DATA SETS
/* *****
/* *
/* * RECEIVE THE MCS CTL STMTS AND SYSMODS
/* *
/* *   EXPECTED RETURN CODE: 0000
/* *
//RECEIVE EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
//        DISP=SHR
//SMPSCDS DD DSN=BUILDER.R140.SMPSCDS,
//        DISP=SHR
//SMPSTS DD DSN=BUILDER.R140.SMPSTS,
//        DISP=SHR
//SMPMTS DD DSN=BUILDER.R140.SMPMTS,
//        DISP=SHR
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,
//        DISP=SHR
//SMPLOG DD DSN=BUILDER.R140.SMPLOG,
//        DISP=SHR
```

---

### BLSMPE#3 (cont.)

```
//SMPLOGA DD DSN=BUILDER.R140.SMPLOGA,
// DISP=SHR
//SMPPTFIN DD DSN=BUILDER.R140.PREP.JCLCNTL(BLSMCS#0),
// DISP=SHR
// DD DSN=BUILDER.R140.PREP.JCLCNTL(BLSMCS#1),
// DISP=SHR
// DD DSN=BUILDER.R140.PREP.JCLCNTL(BLSMCS#2),
// DISP=SHR
// DD DSN=BUILDER.R140.PREP.JCLCNTL(BLSMCS#3),
// DISP=SHR
// DD DSN=BUILDER.R140.PREP.JCLCNTL(BLSMCS#4),
// DISP=SHR
//SMPCNTL DD *
// SET BDY(GLOBAL).
// RECEIVE SYSMODS LIST.
// LIST.
//
/*
//
```

### BLSMPE#4

```
//BLSMPE#4 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLSMPE#4 - RECEIVE THE PTF AND APAR SYSMODS INTO THE
//* - GLOBAL ZONE/DATA SETS
//*****
//*
//* RECEIVE THE PTF AND APAR SYSMODS
//*
//* EXPECTED RETURN CODE: 0000
//*
//RECEIVE EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
// DISP=SHR
//SMPSCDS DD DSN=BUILDER.R140.SMPSCDS,
// DISP=SHR
//SMPSTS DD DSN=BUILDER.R140.SMPSTS,
// DISP=SHR
//SMPMTS DD DSN=BUILDER.R140.SMPMTS,
// DISP=SHR
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,
// DISP=SHR
//SMPLOG DD DSN=BUILDER.R140.SMPLOG,
// DISP=SHR
//SMPLOGA DD DSN=BUILDER.R140.SMPLOGA,
// DISP=SHR
//SMPPTFIN DD DSN=BUILDER.R140.PREP.JCLCNTL(PTFS),
// DISP=SHR
// DD DSN=BUILDER.R140.PREP.JCLCNTL(APARS),
// DISP=SHR
//SMPCNTL DD *
// SET BDY(GLOBAL).
// RECEIVE SYSMODS LIST.
// LIST.
//
/*
//
```

---

## BLSMPE#5

```
//BLSMPE#5 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*****
//* BLSMPE#5 - APPLY THE ELEMENTS (MODULES) INTO THE
//*           - INTO TARGET ZONE/LIBRARIES
//*****
//*
//* APPLY SYSMODS (MODULES)
//*
//*   EXPECTED RETURN CODE: 0000
//*
//APPLY EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
//        DISP=SHR
//SMPSCDS DD DSN=BUILDER.R140.SMPSCDS,
//        DISP=SHR
//SMPSTS DD DSN=BUILDER.R140.SMPSTS,
//        DISP=SHR
//SMPMTS DD DSN=BUILDER.R140.SMPMTS,
//        DISP=SHR
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,
//        DISP=SHR
//IJCLIN DD DSN=BUILDER.R140.PREP.JCLCNTL,
//        DISP=SHR
//BLILOAD DD DSN=BUILDER.R140.SMPE.I.BLOAD,
//        DISP=SHR
//BLISAMP DD DSN=BUILDER.R140.SMPE.I.BLSAMP,
//        DISP=SHR
//CLILOAD DD DSN=BUILDER.R140.SMPE.I.CLLOAD,
//        DISP=SHR
//WBILOAD DD DSN=BUILDER.R140.SMPE.I.WBLOAD,
//        DISP=SHR
//WBICLIST DD DSN=BUILDER.R140.SMPE.I.WBCLIST,
//        DISP=SHR
//WBIPANEL DD DSN=BUILDER.R140.SMPE.I.WBPANEL,
//        DISP=SHR
//WBIMSGS DD DSN=BUILDER.R140.SMPE.I.WBMSGs,
//        DISP=SHR
//WBSKELS DD DSN=BUILDER.R140.SMPE.I.WBSKELS,
//        DISP=SHR
//SCILINK DD DSN=BUILDER.R140.SMPE.I.SCLINK,
//        DISP=SHR
//*
//SMPCNTL DD *
//        SET BDY(BL140TZ).
//        APPLY SELECT(CCVCL140).
//        LIST.
//*
```

## BLSMPE#6

```
//BLSMPE#6 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*****
//* BLSMPE#6 - APPLY THE PTFS INTO THE TARGET ZONE/LIBRARIES
//*           - USING AN IN-STREAM PROCEDURE EXECUTED ONCE PER PTF
//*
```

---

## BLSMPE#6 (cont.)

```
//*****  
//*  
//* APPLY SYSMODS (PTFS) INSTREAM PROCEDURE  
//*  
//APPLYPF PROC  
//*  
//* APPLY A PTF - EXPECTED RETURN CODE: 0000  
//*  
//APPLY EXEC PGM=GIMSMP,REGION=4M  
//SMPCSI DD DSN=BUILDER.R140.CSI,  
// DISP=SHR  
//SMPCSDS DD DSN=BUILDER.R140.SMPSCDS,  
// DISP=SHR  
//SMPSTS DD DSN=BUILDER.R140.SMPSTS,  
// DISP=SHR  
//SMPMTS DD DSN=BUILDER.R140.SMPMTS,  
// DISP=SHR  
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,  
// DISP=SHR  
//  
// PEND  
//*  
//BL00200 EXEC APPLYPF  
//SMPCNTL DD *  
// SET BDY(BL140TZ).  
// APPLY  
// SELECT (BL00200).  
//*  
//
```

## BLSMPE#7

```
//BLSMPE#7 JOB (ACCT)  
//*  
//* DEFAULT JCL  
//*  
//* BUILT BY THE INSTALLATION PREPARATION DIALOG  
//*  
//*  
//*****  
//* BLSMPE#7 - INSTALLATION VERIFICATION PROCESS RUN  
//* - THIS JOB STREAM IS USED TO DEMONSTRATE TO THE  
//* - INSTALLER THAT THE STANDARD VISION: BUILDER  
//* - INSTALLATION WAS SUCCESSFUL. SEVERAL DIFFERENT  
//* - JOB STEPS ARE RUN TO PERFORM A VARIETY OF FUNCTIONS  
//* - INDICATING THAT THE STANDARD PRODUCT IS OPERATIONAL.  
//*  
//* *** NO PERMANENT DATA SETS ARE CREATED BY THIS JOB STREAM ***  
//* *** THE TEMPORARY DATA SETS USE "UNIT=SYSDA" ***  
//*  
//*****  
//*  
//JOB LIB DD DSN=BUILDER.R140.SMPE.T.BLSYSL,  
// DISP=SHR  
//*  
//* INITIALIZE AN M4LIB  
//*  
//INIT EXEC PGM=MARKINIT,REGION=1M  
//M4LIST DD SYSOUT=*  
//M4LIB DD DSN=&&TEMPLIB,DISP=(,PASS),  
// UNIT=SYSDA,  
// SPACE=(TRK,2,,CONTIG)  
//*  
//* CATALOG TABLE AND FILE DEFINITIONS  
//*  
//DEFRUN1 EXEC PGM=MARKIV,REGION=2M  
//M4LIB DD DSN=&&TEMPLIB,DISP=(OLD,PASS)  
//M4LIST DD SYSOUT=*  
//M4INPUT DD *  
DEFRUN RC
```

## BLSMPE#7 (cont.)

```
STATETABTBSR 2C 10C
STATETABTE CA CALIFORNIA
STATETABTE GA GEORGIA
STATETABTE HA HAWAII
STATETABTE NY NEW YORK
CUSTFILEFD RBF 80
CUSTFILELSSEGS10 10
CUSTFILEL0CUSTNUM 101 1 6C1
CUSTFILEL1CUSTNUM CUSTOMER
CUSTFILEL2CUSTNUM NUMBER
CUSTFILELXCUSTNUM CUSTOMER_NUMBER
CUSTFILEL0CUSTNAME 101 7 20C
CUSTFILEL1CUSTNAME CUSTOMER
CUSTFILEL2CUSTNAME NAME
CUSTFILELXCUSTNAME CUSTOMER_NAME
CUSTFILEL0CINDSTRE 101 27 15C
CUSTFILEL1CINDSTRE CUSTOMER
CUSTFILEL2CINDSTRE INDUSTRY
CUSTFILELXCINDSTRE CUSTOMER_INDUSTY
CUSTFILEL0CSTCODE 101 42 2C
CUSTFILEL1CSTCODE CUSTOMER
CUSTFILEL2CSTCODE STATE CODE
CUSTFILELXCSTCODE CUSTOMER_STATE_CODE
/*
/**
/** PROCESSING - SINGLE STEP - LIST MASTER FILE DATA
/**
//PROCRUN EXEC PGM=MARKIV,REGION=2M
//M4LIST DD SYSOUT=*
//M4REPO DD DISP=(NEW,PASS),UNIT=SYSDA,
// SPACE=(TRK,(5,2),RLSE)
//M4SORT DD DISP=(NEW,PASS),UNIT=SYSDA,
// SPACE=(TRK,1)
//SYSOUT DD SYSOUT=*
//SORTWK01 DD SPACE=(CYL,1,,CONTIG),UNIT=SYSDA
//SORTWK02 DD SPACE=(CYL,1,,CONTIG),UNIT=SYSDA
//SORTWK03 DD SPACE=(CYL,1,,CONTIG),UNIT=SYSDA
//M4LIB DD DSN=&&TEMPLIB,DISP=(OLD,PASS)
//M4INPUT DD *
CONTROL
FILE MASTER INPUT NAME CUSTFILE
FILE REPORT
PROC
;
STATE: FIELD C 10 HEADING 'STATE' 'DESCRIPTION'
;
LET T.STATE = LOOKUP(STATETAB CUSTOMER_STATE_CODE)
;
REPORT CUSTOMER NUMBER, CUSTOMER NAME,
CUSTOMER_INDUSTY, CUSTOMER_STATE_CODE,
T.STATE
TITLE 'LIST THE MASTER FILE DATA BY CUSTOMER NAME'
ORDER BY CUSTOMER_NAME
;
END REPORT
;
REPORT CUSTOMER NUMBER, CUSTOMER NAME,
CUSTOMER_INDUSTY, CUSTOMER_STATE_CODE,
T.STATE
TITLE 'LIST THE MASTER FILE DATA BY STATE CODE'
ORDER BY CUSTOMER_STATE_CODE
;
END REPORT
;
END PROC
/*
//M4OLD DD *
001000ABC INSURANCE CO INSURANCE GA
002000THE MONEY CO FINANCIAL NY
003100EVERYTHING GOES RECREATION CA
005100LOGICAL SOFTWARE SOFTWARE NY
```

---

### BLSMPE#7 (cont.)

```
031510FLOATATION INC      MANUFACTURING HA
095100ON THE GO .COM      TRANSPORTATION CA
/*
//*
//* DUMP, INIT, RESTORE AN M4LIB
//*
//LIBRUN EXEC PGM=MARKUTIL,REGION=2M
//M4LIST DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//ABNLIGNR DD DUMMY
//M4LIB DD DSN= &&TEMPLIB,DISP=(OLD,PASS)
//M4WORK DD DSN= &&M4WORK,DISP=(,PASS),
//      SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA
//M4INPUT DD *
//      UCDUMP
//      UCINIT
//      UCREST
/*
//*
//* LIST FILE AND TABLE DEFINITION GLOSSARIES
//*
//DEFRUN2 EXEC PGM=MARKIV,REGION=2M
//M4LIB DD DSN= &&TEMPLIB,DISP=(OLD,PASS)
//M4LIST DD SYSOUT=*
//M4INPUT DD *
LISTGLOSRC
//      CTLEFGCUSTFILE
//      CTLTGSTATETAB
/*
//
```

### BLSMPE#8

```
//BLSMPE#8 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLSMPE#8 - ACCEPT THE ELEMENTS (MODULES) INTO THE
//* - DISTRIBUTION ZONE/LIBRARIES
//*****
//*
//* ACCEPT SYSMODS (MODULES)
//*
//* EXPECTED RETURN CODE: 0000
//*
//ACCEPT EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
//      DISP=SHR
//SMPSCDS DD DSN=BUILDER.R140.SMPSCDS,
//      DISP=SHR
//SMPSTS DD DSN=BUILDER.R140.SMPSTS,
//      DISP=SHR
//SMPMTS DD DSN=BUILDER.R140.SMPMTS,
//      DISP=SHR
//SMPPTS DD DSN=BUILDER.R140.SMPPTS,
//      DISP=SHR
//IJCLIN DD DSN=BUILDER.R140.PREP.JCLCNTL,
//      DISP=SHR
//BLILOAD DD DSN=BUILDER.R140.SMPE.I.BLOAD,
//      DISP=SHR
//BLISAMP DD DSN=BUILDER.R140.SMPE.I.BLSAMP,
//      DISP=SHR
//CLILOAD DD DSN=BUILDER.R140.SMPE.I.CLLOAD,
//      DISP=SHR
//WBILOAD DD DSN=BUILDER.R140.SMPE.I.WBLOAD,
```

---

### BLSMPE#8 (cont.)

```
//          DISP=SHR
//WBICLIST DD DSN=BUILDER.R140.SMPE.I.WBCLIST,
//          DISP=SHR
//WBIPANEL DD DSN=BUILDER.R140.SMPE.I.WBPANEL,
//          DISP=SHR
//WBIMSGS  DD DSN=BUILDER.R140.SMPE.I.WBMSGs,
//          DISP=SHR
//WBISKELS DD DSN=BUILDER.R140.SMPE.I.WBSKELS,
//          DISP=SHR
//SCILINK  DD DSN=BUILDER.R140.SMPE.I.SCLINK,
//          DISP=SHR
//*
//SMPCNTL  DD *
//          SET BDY(BL140DZ).
//          ACCEPT SELECT(CCV140).
//          LIST.
//
//
//
```

### BLSMPE#9

```
//BLSMPE#9 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLSMPE#9 - ACCEPT THE PTFS INTO THE DISTRIBUTION ZONE/LIBRARIES
//*           - USING AN IN-STREAM PROCEDURE EXECUTED ONCE PER PTF
//*****
//*
//* ACCEPT SYSMODS (PTFS) INSTREAM PROCEDURE
//*
//ACCEPTF PROC
//*
//* ACCEPT A PTF - EXPECTED RETURN CODE: 0000
//*
//ACCEPT EXEC PGM=GIMSMP,REGION=4M
//SMPCSI DD DSN=BUILDER.R140.CSI,
//        DISP=SHR
//SMFSCDS DD DSN=BUILDER.R140.SMFSCDS,
//        DISP=SHR
//SMFSTS DD DSN=BUILDER.R140.SMFSTS,
//        DISP=SHR
//SMFMTS DD DSN=BUILDER.R140.SMFMTS,
//        DISP=SHR
//SMFPPTS DD DSN=BUILDER.R140.SMFPPTS,
//        DISP=SHR
//        PEND
//*
//BL00200 EXEC ACCEPTF
//SMPCNTL DD *
//        SET BDY(BL140DZ).
//        ACCEPT
//        SELECT(BL00200).
//
//
//
```

## BLXASM#1

```
//BLXASM#1 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXASM#1 - ASSEMBLE AND LINK THE "PARAMETER" MODULES
//*           - M4PARAMS, M4SFARM, M4LEPARM, MARKLIBP.
//*****
//*
//* THIS INSTREAM PROCEDURE AND JOB STREAM ASSEMBLES AND LINKS THE
//* VISION:UILDER SYSTEM AND COMPONENT PARAMETER MODULES
//* M4PARAMS, M4SFARM, M4LEPARM, MARKLIBP.
//*
//ASMBPRM PROC SRCLIB=,
//              SRCMEM=,
//              BLOAD=
//ASM          EXEC PGM=ASMA90,REGION=2M,
//              PARM='NODECK,OBJECT,LIST'
//SYSLIB       DD DSN=SYS1.MACLIB,
//              DISP=SHR
//SYSUT1       DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIN       DD DSN=&&OBJECT,DISP=(,PASS),UNIT=SYSDA,
//              SPACE=(TRK,(1,1))
//SYSPRINT     DD SYSOUT=*
//SYSIN        DD DSN=&SRCLIB(&SRCMEM),DISP=SHR
//*
//LINK         EXEC PGM=HEWL,REGION=2M,
//              PARM='LET,LIST,MAP,NCAL'
//SYSUT1       DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//OBJECT       DD DSN=&&OBJECT,DISP=(OLD,DELETE)
//SYSLMOD      DD DSN=&BLOAD,DISP=SHR
//LOADLIB      DD DSN=&BLOAD,DISP=SHR
//SYSPRINT     DD SYSOUT=*
//              PEND
//*
//*****
//* THE FOLLOWING IS A SAMPLE EXECUTION OF THE JOBS TO ASSEMBLE
//* AND LINK THE BUILDER SYSTEM AND COMPONENT PARAMETER MODULES.
//*
//* BEFORE YOU RUN THIS JOB, SPECIFY:
//*
//* SRCLIB - THE SOURCE PDS THAT CONTAINS THE PARAMETER MODULE.
//* SRCMEM - THE MEMBER NAME OF THE PARAMETER MODULE.
//* BLOAD - THE LOAD LIBRARY TO CONTAIN THE LINK EDITED MODULE.
//*
//*****
//*
//* M4PARAMS ASSEMBLY AND LINK
//*
//M4PARAM EXEC ASMBPRM,
//              SRCLIB='BUILDER.R140.SMPE.T.BLSAMP',
//              SRCMEM='M4PARAMS',
//              BLOAD='BUILDER.R140.SMPE.T.BLSYSL'
//*
//LINK.SYSLIN DD *
//              INCLUDE OBJECT
//              NAME M4PARAMS(R)
//*
//*
//*****
//* M4SFARM ASSEMBLY AND LINK
//*
//M4SFARM EXEC ASMBPRM,
```

---

## BLXASM#1 (cont.)

```
//          SRCLIB='BUILDER.R140.SMPE.T.BLSAMP',
//          SRCMEM='M4SFARM',
//          BLOAD='BUILDER.R140.SMPE.T.BLSYSL'
// *
//LINK.SYSLIN DD *
//  INCLUDE OBJECT
//    NAME M4SFARM(R)
// *
// *
// *****
// *
// * M4LEPARG ASSEMBLY AND LINK
// *
//M4LEPRM EXEC ASMBPRM,
//          SRCLIB='BUILDER.R140.SMPE.T.BLSAMP',
//          SRCMEM='M4LEPARG',
//          BLOAD='BUILDER.R140.SMPE.T.BLSYSL'
// *
//LINK.SYSLIN DD *
//  INCLUDE OBJECT
//    NAME M4LEPARG(R)
// *
// *
// *****
// *
// * MARKLIBP ASSEMBLY AND LINK
// *
//CLPARG EXEC ASMBPRM,
//          SRCLIB='BUILDER.R140.SMPE.T.BLSAMP',
//          SRCMEM='MARKLIBP',
//          BLOAD='BUILDER.R140.SMPE.T.BLSYSL'
// *
//LINK.SYSLIN DD *
//  INCLUDE OBJECT
//    NAME MARKLIBP(R)
// *
//
```

## BLXASM#2

```
//BLXASM#2 JOB (ACCT)
// *
// * DEFAULT JCL
// *
// * BUILT BY THE INSTALLATION PREPARATION DIALOG
// *
// *
// *****
// * BLXASM#2 - ASSEMBLE AND LINK THE "PARG" MODULES
// *           - OQLPARG AND BQLPARG.
// *
// *****
// * THIS INSTREAM PROCEDURE AND JOB STREAM ASSEMBLES AND LINKS THE
// * VISION:BUILDER SYSTEM AND COMPONENT PARG MODULES
// * OQLPARG AND BQLPARG.
// *
//ASMBPRM PROC SRCLIB=,
//             SRCMEM=,
//             BLOAD=
//ASM EXEC PGM=ASMA90,REGION=2M,
//          PARG='NODECK,OBJECT,LIST'
//SYSLIB DD DSN=SYS1.MACLIB,
//          DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIN DD DSN=&OBJECT,DISP=(,PASS),UNIT=SYSDA,
//          SPACE=(TRK,(1,1))
//SYSPRINT DD SYSOUT=*
//SYSIN DD DSN=&SRCLIB(&SRCMEM),DISP=SHR
// *
```

---

## BLXASM#2 (cont.)

```
//LINK      EXEC PGM=HEWL,REGION=2M,
//          PARM='LET,LIST,MAP,NCAL'
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//OBJECT     DD DSN=&&OBJECT,DISP=(OLD,DELETE)
//SYSLMOD    DD DSN=&BLLOAD,DISP=SHR
//LOADLIB    DD DSN=&BLLOAD,DISP=SHR
//SYSPRINT   DD SYSOUT=*
//          PEND
//*
//*****
//*
//* THE FOLLOWING IS A SAMPLE EXECUTION OF THE JOBS TO ASSEMBLE
//* AND LINK THE BUILDER SYSTEM AND COMPONENT PARM MODULES.
//*
//* BEFORE YOU RUN THIS JOB, SPECIFY:
//*
//* SRCLIB - THE SOURCE PDS THAT CONTAINS THE PARM MODULE.
//* SRCMEM - THE MEMBER NAME OF THE PARM MODULE.
//* BLLOAD - THE LOAD LIBRARY TO CONTAIN THE LINK EDITED MODULE.
//*
//*****
//*
//* OQLPARM ASSEMBLY AND LINK
//*
//OQLPARM EXEC ASMBPRM,
//          SRCLIB='BUILDER.R140.SMPE.T.BLSAMP',
//          SRCMEM='OQLPARM',
//          BLLOAD='BUILDER.R140.SMPE.T.BLSYSL'
//*
//LINK.SYSLIN DD *
//          REPLACE OQLPARM
//          INCLUDE LOADLIB(OQL)
//          INCLUDE OBJECT
//          ENTRY QLMOQL
//          ALIAS QUERYIV
//          NAME OQL(R)
//*
//*
//*****
//*
//* BQLPARM ASSEMBLY AND LINK
//*
//BQLPARM EXEC ASMBPRM,
//          SRCLIB='BUILDER.R140.SMPE.T.BLSAMP',
//          SRCMEM='BQLPARM',
//          BLLOAD='BUILDER.R140.SMPE.T.BLSYSL'
//*
//LINK.SYSLIN DD *
//          REPLACE BQLPARM
//          INCLUDE LOADLIB(BQL)
//          INCLUDE OBJECT
//          ENTRY QLMBQL
//          NAME BQL(R)
//*
//
```

## BLXBAN#1

```
//BLXBAN#1 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXBAN#1 - A VISION:BUILDER RUN TO OUTPUT A SIGNON BANNER PAGE
//*****
//*
```

## BLXBAN#1 (cont.)

```
/* THIS JOB STREAM EXECUTES A NULL DEFINITION RUN IN ORDER TO
/* PRODUCE THE VISION:UILDER SIGNON BANNER PAGE.
/*
/* BEFORE YOU RUN THIS JOB, REVIEW THE JCL AND SPECIFY:
/*
/* JOBLIB - THE VISION:UILDER SYSTEM LOAD LIBRARY
/*
/* JOBLIB DD DSN=UILDER.R140.SMPE.T.BLSYSL
/*
/* BANNER EXEC PGM=MARKIV,REGION=2M
/* M4LIST DD SYSOUT=*
/* M4LIB DD DUMMY
/* M4INPUT DD *
BANNER RC
/*
/*
```

## BLXCBQ#1

```
/*BLXCBQ#1 JOB (ACCT)
/*
/* DEFAULT JCL
/*
/* BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/*
/* *****
/* BLXCBQ#1 - LINK THE COBOL QUICK START UTILITY
/* - WITH THE CA-LIBRARIAN INTERFACE MODULES
/* *****
/*
/* CLLIBLK PROC BLOAD=,
/* LIBLOAD=
/* LINK EXEC PGM=HEWL,REGION=2M,PARM='LET,LIST,MAP,NCAL'
/* SYSLIB DD DUMMY
/* SYSPRINT DD SYSOUT=*
/* SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
/* LIBSYS DD DSN=&LIBLOAD,
/* DISP=SHR
/* LLIB DD DSN=&BLOAD,
/* DISP=SHR
/* SYSLMOD DD DSN=&BLOAD,
/* DISP=SHR
/* PEND
/*
/* *****
/* BEFORE SUBMITTING THIS JCL, YOU MUST SPECIFY THE FOLLOWING
/* INFORMATION:
/*
/* BLOAD - NAME OF THE VISION:UILDER SYSTEM LOADLIB.
/* LIBLOAD - NAME OF YOUR CA-LIBRARIAN SYSTEM LOADLIB.
/*
/* NOTE: A CONDITION CODE OF 4 FROM THE LINK EDIT RUN IS OKAY.
/*
/* *****
/*
/* LIBLINK EXEC CLLIBLK,
/* BLOAD='UILDER.R140.SMPE.T.BLSYSL',
/* LIBLOAD='LIBRARIAN.LOADLIB'
/* LINK.SYSLIN DD *
INCLUDE LIBSYS (FAIRCLS)
INCLUDE LIBSYS (FAIRERR)
INCLUDE LIBSYS (FAIRLOC)
INCLUDE LIBSYS (FAIRMOD)
INCLUDE LIBSYS (FAIRNTE)
INCLUDE LIBSYS (FAIROPN)
INCLUDE LIBSYS (FAIRPNT)
INCLUDE LIBSYS (FAIRREC)
```

---

### BLXCBQ#1 (cont.)

```
INCLUDE LIBSYS (FAIRSEC)
INCLUDE LLIB (COMLIBL)
ENTRY COMLIBL
NAME COMLIBL (R)
/*
//
```

### BLXCBQ#2

```
//BLXCBQ#2 JOB (ACCT)
//*
/* DEFAULT JCL
/*
/* BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/*
/******
/* BLXCBQ#2 - LINK THE COBOL QUICK START UTILITY
/*           - WITH THE CA-PANVALET INTERFACE MODULES
/******
/*
//CLPANLK PROC BLOAD=,
//              PANLOAD=
//LINK EXEC PGM=IEWL,REGION=2M,PARM='LET,LIST,MAP,NCAL'
//SYSLIB DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//LIBSYS DD DSN=&PANLOAD,
//          DISP=SHR
//LLIB DD DSN=&BLOAD,
//          DISP=SHR
//SYSIMOD DD DSN=&BLOAD,
//          DISP=SHR
//          PEND
//
/******
/* BEFORE SUBMITTING THIS JCL, YOU MUST SPECIFY THE FOLLOWING
/* INFORMATION:
/*
/* BLOAD - NAME OF THE VISION:BUILDER SYSTEM LOADLIB.
/* PANLOAD - NAME OF YOUR CA-PANVALET SYSTEM LOADLIB.
/*
/******
/*
//PANLINK EXEC CLPANLK,
//              BLOAD='BUILDER.R140.SMPE.T.BLSYSL',
//              PANLOAD='PANVALET.SYSTEM.LOADLIB'
//LINK.SYSLIN DD *
//              INCLUDE LIBSYS (PAM)
//              INCLUDE LLIB (COMLIBP)
//              ENTRY COMLIBP
//              NAME COMLIBP (R)
/*
//
```

### BLXCBQ#3

```
//BLXCBQ#3 JOB (ACCT)
//*
/* DEFAULT JCL
/*
/* BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/*
/******
/* BLXCBQ#3 - RUN THE COBOL QUICK START UTILITY
/******
```

### BLXCBQ#3 (cont.)

```
/*
/* ***** NOTE *****
/* THE SYSCOPY DD STATEMENT IS USED FOR MVS COPYBOOK LIBRARIES.
/* THE PANDD1 DD STATEMENT IS USED FOR PANVALET COPYBOOK LIBRARIES.
/* THE MASTER DD STATEMENT IS USED FOR LIBRARIAN COPYBOOK LIBRARIES
/*
/******
/*
//COBOLQS PROC BLOAD=,
//      COPYLIB=,
//      DEFLIB=,
//      MEMBER=
//COBOLQS EXEC PGM=COBOLQS,REGION=2M
//STEPLIB DD DSN=&BLOAD,
//      DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSCOPY DD DSN=&COPYLIB,
//      DISP=SHR
//PANDD1 DD DSN=&COPYLIB,
//      DISP=SHR
//MASTER DD DSN=&COPYLIB,
//      DISP=SHR
//SYS004 DD DSN=&DEFLIB(&MEMBER),
//      DISP=SHR
//SYSIN DD DUMMY
//      PEND
/*
/******
/* BEFORE SUBMITTING THIS JCL, YOU MUST SPECIFY THE FOLLOWING
/* INFORMATION:
/*
/* BLOAD - NAME OF THE VISION:UILDER SYSTEM LOADLIB
/* COPYLIB - NAME OF YOUR COBOL COPY LIBRARY. THIS IS AN
/*           MVS, PANVALET, OR LIBRARIAN COPYBOOK LIBRARY.
/* DEFLIB - NAME OF YOUR COMLIB SOURCE DEFINITION LIBRARY.
/*           THE GENERATED FILE DEFINITION IS WRITTEN TO
/*           THIS LIBRARY.
/* MEMBER - MEMBER NAME FOR THE DEFINITION YOU ARE GENERATING.
/*
/* YOU MUST ALSO PROVIDE THE APPROPRIATE SYSIN DATA IN THE
/* COBOLQS.SYSIN DD OVERRIDE STMT.
/******
/*
//QS EXEC COBOLQS,
//      BLOAD='BUILDER.R140.SMPE.T.BLSYSL',
//      COPYLIB='COBOL.COPYBOOK.LIBRARY',
//      DEFLIB='YOUR.DEFINITION.OUT.LIB',
//      MEMBER='SAMPLEFD'
//COBOLQS.SYSIN DD *
FILEGEN NAME=SAMPLEFD,TYPE=FIXED,RECSIZE=80
SEGMENT NAME=OFFICE,NUMBER=10,LEVEL=1
$COBOL
    01 OFFICE-DATA.
        02 OFFICE-CODE PIC S9(3).
        02 OFFICE-ADDRESS.
            03 OFFICE-STREET PIC X(20).
            03 OFFICE-CITY PIC X(15).
            03 OFFICE-STATE PIC X(2).
            03 OFFICE-ZIP.
                04 OFFICE-ZIP-FIRST-FIVE PIC X(5).
                04 OFFICE-ZIP-LAST-FOUR PIC X(4).
            02 OFFICE-PHONE PIC 9(7).
            02 OFFICE-AREA-CODE PIC X(3).
            02 SPEED-DIAL PIC X(3).
            02 FILLER PIC X(18).
$ECOBOL
/*
//
```

## BLXCOP#1

```
//BLXCOP#1 JOB (ACCT)
//*
//* DEFAULT JCL
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXCOP#1 - COPY THE TARGET LOAD LIBRARY
//*           - TO A NEW USER LOAD LIBRARY
//*****
//*
//* THIS INSTREAM PROCEDURE WILL COPY THE ENTIRE TARGET LOAD LIBRARY
//* (THE BLSYSL) TO A "NEW" USER LOAD LIBRARY.
//*
//* THE "NEW" USER LOAD LIBRARY IS DELETED FIRST, THEN RE-ALLOCATED.
//*
//*
//COPYLIB PROC TARGETL=,
//                USERLIB=,
//                USERUNT=
//*
//DELETE EXEC PGM=IEFBR14,REGION=120K
//OLD      DD DSN=&USERLIB,
//          DISP=(MOD,DELETE),
//          SPACE=(TRK,(0,0)),
//          UNIT=&USERUNT
//*
//ALLOC EXEC PGM=IEFBR14,REGION=120K
//NEW      DD DSN=&USERLIB,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=USERUNT,
//          SPACE=(TRK,(285,15,120)),
//          DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)
//*
//COPY EXEC PGM=IEBCOPY,REGION=2M
//SYSPRINT DD SYSOUT=*
//OUT      DD DSN=&USERLIB,
//          DISP=OLD
//IN        DD DSN=&TARGETL,
//          DISP=SHR
//SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,60)
//SYSUT4    DD UNIT=SYSDA,SPACE=(TRK,60)
//*
//          PEND
//*
//*****
//*
//* THE FOLLOWING IS A SAMPLE EXECUTION OF THE JOB TO COPY THE
//* ENTIRE TARGET LOAD LIBRARY TO A NEW USER LOAD LIBRARY.
//*
//* BEFORE YOU RUN THIS JOB, SPECIFY:
//*
//* TARGETL - THE TARGET LOAD LIBRARY NAME (BLSYSL)
//* USERLIB - THE "NEW" USER LOAD LIBRARY NAME
//* USERUNT - THE USER LOAD LIBRARY UNIT
//*
//*****
//* COPY THE TARGET LOAD LIBRARY
//*
//COPYLIB EXEC COPYLIB,
//          TARGETL='BUILDER.R140.SMPE.T.BLSYSL',
//          USERLIB='THE NEW USER LOADLIB',
```

---

## BLXCOP#1 (cont.)

```
//          USERUNT='SYSDA'
// *
// COPY.SYSIN      DD *
//   COPY INDD=IN,OUTDD=OUT
// *
//
```

## BLXCOP#2

```
//BLXCOP#2 JOB (ACCT)
// *
// * DEFAULT JCL
// *
// * BUILT BY THE INSTALLATION PREPARATION DIALOG
// *
// *
// *****
// * BLXCOP#2 - COPY (WITH REPLACE) THE TARGET LOAD LIBRARY
// *           - TO AN EXISTING USER LOAD LIBRARY
// *
// *****
// * THIS INSTREAM PROCEDURE WILL COPY THE ENTIRE TARGET LOAD LIBRARY
// * (THE BLSYSL) TO AN "EXISTING" USER LOAD LIBRARY.
// *
// * THE "EXISTING" USER LOAD LIBRARY MEMBERS ARE REPLACED.
// *
// *
// COPYLIB PROC TARGETL=,
//           USERLIB=
// *
// COPY      EXEC PGM=IEBCOPY,REGION=2M
// SYSPRINT  DD SYSOUT=*
// OUT       DD DSN=USERLIB,
//           DISP=OLD
// IN        DD DSN=TARGETL,
//           DISP=SHR
// SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,150)
// SYSUT4    DD UNIT=SYSDA,SPACE=(TRK,150)
// *
//           PEND
// *
// *****
// *
// * THE FOLLOWING IS A SAMPLE EXECUTION OF THE JOB TO COPY THE
// * ENTIRE TARGET LOAD LIBRARY TO AN EXISTING USER LOAD LIBRARY.
// *
// * BEFORE YOU RUN THIS JOB, SPECIFY:
// *
// * TARGETL - THE TARGET LOAD LIBRARY NAME (BLSYSL)
// * USERLIB - THE "EXISTING" USER LOAD LIBRARY NAME
// *
// *****
// *
// * COPY THE TARGET LOAD LIBRARY
// *
// COPYLIB EXEC COPYLIB,
//           TARGETL='BUILDER.R140.SMPE.T.BLSYSL',
//           USERLIB='THE EXISTING USER LOADLIB'
// *
// COPY.SYSIN      DD *
//   COPY INDD=((IN,R)),OUTDD=OUT
//   COPY INDD=OUT,OUTDD=OUT
// *
//
```

```

//BLXDBQ#1 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXDBQ#1 - BIND THE DB2 QUICK START UTILITY
//*****
//*
//* BIND THE DB2 PLAN FOR THE DB2 QUICK START UTILITY.
//*
//* BEFORE SUBMITTING THIS JCL, YOU MUST SPECIFY THE FOLLOWING
//* INFORMATION:
//*
//* - NAME OF YOUR DB2 SYSTEM DSNLOAD LIBRARY AS THE
//* STEPLIB DATA SET NAME.
//*
//* - APPROPRIATE VALUES FOR THE SYSTSIN DSN PARAMETERS:
//*
//* PARAMETER
//* -----
//* SYSTEM NAME OF YOUR DB2 SUBSYSTEM ID.
//* PLAN NAME OF YOUR DB2 PLAN FOR THE DB2 QUICK
//* START UTILITY.
//* LIBRARY NAME OF THE LIBRARY CONTAINING THE DB2
//* QUICK START DATA BASE REQUEST MODULE (DBRM).
//* THE DB2 QUICK START DBRM IS DELIVERED IN THE
//* BUILDER WORKLIB PDS MEMBER DB2QDBRM.
//* MEMBER NAME OF THE DB2 QUICK START DBRM - DB2QDBRM
//* QUALIFIER NAME OF THE QUALIFIER FOR YOUR DB2 SYSTEM
//* CATALOG TABLE. THIS MUST BE THE QUALIFIER
//* FOR YOUR SYSCOLUMNS TABLE (E.G., SYSIBM).
//*****
//*
//BINDPGM EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=2M
//STEPLIB DD DSN=DB2.SDSNLOAD,
// DISP=SHR
//SYSTSPT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB2T)
BIND PLAN (DB2QS) -
LIBRARY ('BUILDER.R140.SMPE.T.BLSAMP') -
MEMBER (DB2QDBRM) -
QUALIFIER (SYSIBM) -
ACTION (REPLACE ) RETAIN
END
//
//

```

```
//BLXDBQ#2 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXDBQ#2 - EXECUTE THE DB2 QUICK START UTILITY
//*****
//*
//* EXECUTE THE DB2 QUICK START UTILITY.
//*
//DB2QS PROC BLOAD=,
```

---

## BLXDBQ#2 (cont.)

```
//          DB2LOAD=,
//          DEFLIB=
//DB2QS    EXEC PGM=DB2QS,REGION=2M
//STEPLIB  DD DSN=&BLOAD,DISP=SHR
//          DD DSN=&DB2LOAD,DISP=SHR
//SYSTEM   DD DUMMY
//SYSPRINT DD SYSOUT=*,
//          DCB=(DSORG=PS,RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYS004   DD DSN=&DEFLIB,DISP=OLD
//SYSIN    DD DUMMY
//          PEND
//*
//*****
//* BEFORE SUBMITTING THIS JCL, YOU MUST SPECIFY THE FOLLOWING
//* INFORMATION:
//*
//*   BLOAD - NAME OF THE VISION:BUILDER SYSTEM LOADLIB
//*   DB2LOAD - NAME OF YOUR DB2 DSN.DSNLOAD LIBRARY.
//*   DEFLIB - NAME OF YOUR COMLIB SOURCE DEFINITION LIBRARY.
//*           THE GENERATED FILE DEFINITION IS WRITTEN TO
//*           THIS LIBRARY.
//*
//* YOU MUST ALSO PROVIDE THE APPROPRIATE SYSIN DATA IN THE
//* DB2QS.SYSIN DD OVERRIDE STMT.
//*****
//*
//QS        EXEC DB2QS,
//          BLOAD='BUILDER.R140.SMPE.T.BLSYSL',
//          DB2LOAD='DB2.SDSNLOAD',
//          DEFLIB='YOUR.DEFINITION.OUT.LIB'
//*
//DB2QS.SYSIN DD *
DB2CNTL DB2PLAN=DB2QS,DB2SYS=D61A
FILEGEN NAME=DB2FD,BUFFSIZE=1024K
SEGMENT NAME=DEPT,NUMBER=10,LEVEL=1,TABLE=DEPT,CREATOR=DSN8610,
PRINT=ALL
NEWPAGE
SEGMENT NAME=EMPLOYEE,NUMBER=20,LEVEL=2,TABLE=EMP,CREATOR=DSN8610,
PRINT=ALL
NEWPAGE
SEGMENT NAME=PROJECT,NUMBER=30,LEVEL=2,TABLE=PROJ,CREATOR=DSN8610,
PRINT=ALL
//*
//
```

## BLXDB2#T

```
//BLXDB2#T JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*****
//* BLXDB2#T - PREPARE THE "MARKSQL" MODULE
//*           - FOR TERADATA TABLE ACCESS
//*****
//*
//* THIS FOLLOWING SAMPLE JOB STREAM ASSEMBLES, PREPARES AND LINKS
//* THE "MARKSQL" MODULE FOR USE WITH TERADATA DATABASES THROUGH THE
//* STANDARD DB2 FACILITY OF VISION:BUILDER.
//*
//* THE MODULE NAME GENERATED IS "MARKSQLT". SEE THE INSTALLATION
//* INSTRUCTIONS FOR CONSIDERATIONS WHEN GENERATING THIS MODULE.
//*
//* NOTE: THE STANDARD IBM PROCEDURE "DSNHASM" IS USED TO ACCOMPLISH
//*       THE DB2 PREPARATION PROCESS. YOU MAY NEED TO CONFER WITH
//*       YOUR DATA BASE ADMINISTRATOR BEFORE RUNNING THIS JOB.
```

---

## BLXDB2#T (cont.)

```
//*
//* BEFORE YOU RUN THESE JOBS,
//* REVIEW THE NAMED "JCL" STATEMENTS AND SPECIFY:
//*
//*   SYSIN   - THE DATASET/MEMBER FOR THE "MARKSQL" SOURCE CODE.
//*   SQLPREP - THE "MEM" ENTRY IS THE PLAN NAME FOR THE
//*             DB2 RUN MODULE.  THE DEFAULTS ARE SHOWN.
//*             - THE "USER" ENTRY IS THE AUTHORIZED USERID.
//*             THIS NAME IS USED BY THE IBM PROCEDURE "DSNHASM"
//*             TO CONSTRUCT DEFAULT DATASET NAMES.
//*   SYSLIB  - THE LOAD LIBRARY CONTAINING THE TERADATA MODULES.
//*   SYSLMOD - THE LOAD LIBRARY FOR THE LINKED DB2 RUN MODULES.
//*
//*****
//*
//* PREPARE THE TERADATA ATTACH RUN MODULE.
//*
//*
//GENT      EXEC PGM=ASMA90,PARM='DECK,NOBJECT',REGION=2M
//SYSPRINT  DD SYSOUT=*
//SYSPUNCH  DD DSN=&&GENT,DISP=(MOD,PASS),UNIT=SYSDA,
//            SPACE=(800,(300,300)),
//            DCB=(RECFM=FB,BLKSIZE=3200)
//SYSUT1    DD UNIT=SYSDA,SPACE=(800,(300,300))
//SYSLIB     DD DSN=SYS1.MACLIB,
//            DISP=SHR
//SYSIN      DD DSN=BUILDER.R140.SMPE.T.BLSAMP(MARKSQL),
//            DISP=SHR
//*
//SQLPREP   EXEC DSNHASM,MEM=MARKIV,USER=PUBLIC,
//            PARM.PC='HOST(ASM),STDSQL(86)'
//PC.SYSIN   DD DSN=&&GENT,DISP=(OLD,DELETE)
//LKED.SYSLIB DD DSN=TERADATA.TSAPI.LIBRARY,DISP=OLD
//LKED.SYSLMOD DSN=BUILDER.R140.SMPE.T.BLSYSL,
//            DISP=OLD
//LKED.SYSIN  DD *
//            CHANGE DSNHLI2(DSNHLI)
//            INCLUDE SYSLIB(DSNALI)
//            MODE AMODE(31),RMODE(ANY)
//            NAME MARKSQLT(R)
//*
//
```

## BLXDB2#1

```
//BLXDB2#1 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXDB2#1 - PREPARE THE "MARKSQL" MODULE FOR
//*           - FOR THE VARIOUS DB2 ATTACH FACILITIES
//*****
//*
//* THIS FOLLOWING SAMPLE JOB STREAM ASSEMBLES, PREPARES AND LINKS
//* THE "MARKSQL" MODULE THREE TIMES FOR USE WHEN ATTACHING TO
//* DB2 DURING APPLICATION RUNS. THE FOLLOWING MODULES ARE PRODUCED.
//*
//*   MODULE   - ATTACH FACILITY
//*   -----
//*   MARKSQLC - CALL ATTACH
//*   MARKSQLI - IMS ATTACH
//*   MARKSQLT - TSO ATTACH
//*
//* NOTE: THE STANDARD IBM PROCEDURE "DSNHASM" IS USED TO ACCOMPLISH
//*       THE DB2 PREPARATION PROCESS. YOU MAY NEED TO CONFER WITH
```

## BLXDB2#1 (cont.)

```
/*          YOUR DB2 DATA BASE ADMINISTRATOR BEFORE RUNNING THIS JOB.
/*
/* BEFORE YOU RUN THESE JOBS,
/* REVIEW THE NAMED "JCL" STATEMENTS AND SPECIFY:
/*
/* SYSIN - THE DATASET/MEMBER FOR THE "MARKSQL" SOURCE CODE.
/* SQLPREP - THE "MEM" ENTRY IS THE PLAN NAME FOR THE
/* DB2 RUN MODULE. THE DEFAULTS ARE SHOWN.
/* - THE "USER" ENTRY IS THE AUTHORIZED USERID.
/* THIS NAME IS USED BY THE IBM PROCEDURE "DSNHASM"
/* TO CONSTRUCT DEFAULT DATASET NAMES.
/* SYSLMOD - THE LOAD LIBRARY FOR THE LINKED DB2 RUN MODULES.
/* - THE DEFAULT IS THE SMP/E TARGET SYSTEM LIBRARY.
/* SYSLIB - THE IMS LOAD LIBRARY CONTAINING THE IMS INTERFACE
/* MODULE "DFSLI000" INCLUDED DURING LINK EDIT.
/* (NOTE THAT THE STEP WITH THE SYSLIB DD IS ONLY NEEDED
/* FOR THE IMS ATTACH. IF THE IMS ATTACH FACILITY IS
/* NOT NEEDED, REMOVE THE JCL FOR THIS STEP.)
/*
/******
/*
/* PREPARE THE "DB2 CALL ATTACH" MODULE "MARKSQLC"
/*
/*
/*GENC EXEC PGM=ASMA90,PARM='DECK,NOOBJECT',REGION=2M
/*SYSPRINT DD SYSOUT=*
/*SYSPUNCH DD DSN=&&GENC,DISP=(MOD,PASS),UNIT=SYSDA,
/* SPACE=(800,(300,300)),
/* DCB=(RECFM=FB,BLKSIZE=3200)
/*SYSUT1 DD UNIT=SYSDA,SPACE=(800,(300,300))
/*SYSLIB DD DSN=SYS1.MACLIB,
/* DISP=SHR
/*SYSIN DD DISP=SHR,
/* DSN=BUILDER.R140.SMPE.T.BLSAMP (MARKSQL)
/*
/*SQLPREP EXEC DSNHASM,MEM=MARKDB2,USER=PUBLIC,
/* PARM.PC='HOST(ASM),STDSQL(86)'
/*PC.SYSIN DD DSN=&&GENC,DISP=(OLD,DELETE)
/*LKED.SYSLMOD DD DISP=OLD,
/* DSN=BUILDER.R140.SMPE.T.BLSYSL
/*LKED.SYSIN DD *
/* INCLUDE SYSLIB(DSNALI)
/* MODE AMODE(31),RMODE(ANY)
/* NAME MARKSQLC(R)
/*
/*
/******
/*
/* PREPARE THE "DB2 IMS ATTACH" MODULE "MARKSQLI"
/*
/*
/*GENI EXEC PGM=ASMA90,PARM='DECK,NOOBJECT',REGION=2M
/*SYSPRINT DD SYSOUT=*
/*SYSPUNCH DD DSN=&&GENI,DISP=(MOD,PASS),UNIT=SYSDA,
/* SPACE=(800,(300,300)),
/* DCB=(RECFM=FB,BLKSIZE=3200)
/*SYSUT1 DD UNIT=SYSDA,SPACE=(800,(300,300))
/*SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
/*SYSIN DD DISP=SHR,
/* DSN=BUILDER.R140.SMPE.T.BLSAMP (MARKSQL)
/*
/*SQLPREP EXEC DSNHASM,MEM=MARKDLI,USER=PUBLIC,
/* PARM.PC='HOST(ASM),STDSQL(86)'
/*PC.SYSIN DD DSN=&&GENI,DISP=(OLD,DELETE)
/*LKED.SYSLIB DD DISP=SHR,
/* DSN=IMS.RESLIB
/*LKED.SYSLMOD DD DISP=OLD,
/* DSN=BUILDER.R140.SMPE.T.BLSYSL
/*LKED.SYSIN DD *
/* INCLUDE SYSLIB(DFSLI000)
/* MODE AMODE(31),RMODE(ANY)
```

## BLXDB2#1 (cont.)

```
NAME MARKSQLI(R)
/*
//*
//*****
//*
//* PREPARE THE "DB2 TSO ATTACH" MODULE "MARKSQLT"
//*
//*
//GENT EXEC PGM=ASMA90, PARM='DECK,NOOBJECT',REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DSN=&&GENT,DISP=(MOD,PASS),UNIT=SYSDA,
//          SPACE=(800,(300,300)),
//          DCB=(RECFM=FB,BLKSIZE=3200)
//SYSUT1 DD UNIT=SYSDA,SPACE=(800,(300,300))
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
//SYSIN DD DISP=SHR,
//        DSN=BUILDER.R140.SMPE.T.BLSAMP(MARKSQL)
//*
//SQLPREP EXEC DSNHASM,MEM=MARKIV,USER=PUBLIC,
//            PARM.PC='HOST(ASM),STDSQL(86)'
//PC.SYSIN DD DSN=&&GENT,DISP=(OLD,DELETE)
//LKED.SYSLMOD DD DISP=OLD,
//            DSN=BUILDER.R140.SMPE.T.BLSYSL
//LKED.SYSIN DD *
//          INCLUDE SYSLIB(DSNELI)
//          MODE AMODE(31),RMODE(ANY)
//          NAME MARKSQLT(R)
/*
//
```

## BLXDB2#2

```
//BLXDB2#2 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXDB2#2 - BIND THE PREPARED "MARKSQL" MODULES
//*           - USED FOR ATTACHING TO DB2 DURING PROCESSING RUNS
//*****
//*
//* THE FOLLOWING SAMPLE JOB STREAM DOES THE DB2 BINDS FOR THE
//* THREE "MARKSQL" MODULES THAT ARE USED FOR ATTACHING TO DB2
//* DURING APPLICATION RUNS. THE FOLLOWING ARE THE DEFAULT
//* NAMES USED IN THE PREPARATION RUNS AND REFERENCED HERE.
//*
//* MODULE - PLAN NAME ATTACH FACILITY
//* -----
//* MARKSQLC - MARKDB2 - CALL ATTACH
//* MARKSQLI - MARKDLI - IMS ATTACH
//* MARKSQLT - MARKIV - TSO ATTACH
//*
//*
//* BEFORE RUNNING THIS JOB,
//* CONFER WITH YOUR DB2 DATA BASE ADMINISTRATOR AND
//* REVIEW THE VALUES FOR THE FOLLOWING ENTRIES:
//*
//* SYSTEM - THE DB2 SUBSYSTEM NAME.
//* PLAN - THE PLAN NAME. THE DEFAULT IS SHOWN.
//* LIBRARY - THE DBRM LIBRARY NAME.
//* MEMBER - THE PLAN MEMBER NAME. THE SAME AS THE PLAN NAME.
//*
//*****
//* BIND THE "DB2 CALL ATTACH" PLAN "MARKDB2" FOR MODULE "MARKSQLC"
//*
```

---

## BLXDB2#2 (cont.)

```
//*
//BINDC EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=2M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(SUBSYSTEM)
    BIND PLAN (MARKDB2) -
      LIBRARY ('PUBLIC.DBRMLIB.DATA') -
      MEMBER (MARKDB2) -
      ACTION (REPLACE) RETAIN -
      ISOLATION (CS)
  END
/*
//*
//*****
//*
//* BIND THE "DB2 IMS ATTACH" PLAN "MARKDBI" FOR MODULE "MARKSQLI"
//*
//*
//*
//* BINDI EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=2M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(SUBSYSTEM)
    BIND PLAN (MARKDLI) -
      LIBRARY ('PUBLIC.DBRMLIB.DATA') -
      MEMBER (MARKDLI) -
      ACTION (REPLACE) RETAIN -
      ISOLATION (CS)
  END
/*
//*
//*****
//*
//* BIND THE "DB2 TSO ATTACH" PLAN "MARKIV" FOR MODULE "MARKSQLI"
//*
//*
//*
//* BINDT EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=2M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(SUBSYSTEM)
    BIND PLAN (MARKIV) -
      LIBRARY ('PUBLIC.DBRMLIB.DATA') -
      MEMBER (MARKIV) -
      ACTION (REPLACE) RETAIN -
      ISOLATION (CS)
  END
/*
//
```

## BLXINQ#1

```
//BLXINQ#1 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXINQ#1 - RUN THE INQUIRY QUICK START UTILITY
//*****
//*
//* UTILITY TO CONVERT VISION:INQUIRY FILE DEFINITIONS INTO
//* VISION:BUILDER OR VISION:INFORM FORMAT FILE DEFINITIONS.
//* THE VISION:INQUIRY FILE DEFINITIONS MUST COME FROM AN
//* VISION:INQUIRY UNLOADED SYSTEM DATABASE FILE. SEE YOUR
//* VISION:INQUIRY TECHNICAL REFERENCE MANUAL FOR INFORMATION ON
//* HOW TO CREATE AN UNLOADED COPY OF THE SYSTEM DATABASE.
//*
//* THIS UTILITY MAY ALSO BE INVOKED UNDER TSO/ISPF USING THE
```

---

## BLXINQ#1 (cont.)

```
/* VISION:WORKBENCH FOR ISPF.
/*
/******
/*
//INQRYQS PROC RGN=2M,
//          BLOAD=,
//          ULSYSDB=,
//          DEFLIB=
//INQRYQS EXEC PGM=INQRYQS,REGION=&RGN
//STEPLIB DD DSN=&BLOAD,
//          DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=&ULSYSDB,
//          DISP=SHR
//SYS004 DD DSN=&DEFLIB,
//          DISP=SHR
//          PEND
/*
/******
/* FOLLOWING IS A SAMPLE EXECUTION OF THIS PROCEDURE. BEFORE YOU
/* RUN THIS PROCEDURE, SPECIFY:
/*
/* RGN - THE REGION SIZE; DEFAULT IS 2M.
/* BLOAD - THE VISION:UILDER SYSTEM LOADLIB
/* ULSYSDB - THE UNLOADED VISION:INQUIRY SYSTEM DATABASE FILE.
/* DEFLIB - THE VISION:INFORM DEFINITION LIBRARY.
/******
/*
//STEP01 EXEC INQRYQS,RGN=2M,
//          BLOAD='BUILDER.R140.SMPE.T.BLSYSL',
//          ULSYSDB='VISION.INQUIRY.UNLOADED.SYSDBASE',
//          DEFLIB='YOUR.DEFINITION.OUT.LIB'
//SYSIN DD *
//          FILEGEN NAME=VSHPLANT,FLDPREFX=PLT
//          FILEGEN NAME=SALARIES,FLDPREFX=SAL
/*
//
```

## BLXMSG#1

```
/*BLXMSG#1 JOB (ACCT)
/*
/** DEFAULT JCL
/*
/** BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/******
/** BLXMSG#1 - COPY THE VISION:UILDER MESSAGE MODULES
/** - FOR USE IN LOADING THE SYSTEM LPA
/******
/*
/** THIS JOB COPIES THE BUILDER MESSAGES MODULES TO A LOADLIB
/** THAT IS USED TO LOAD MODULES INTO THE SYSTEM LPA.
/*
/** BEFORE YOU RUN THIS JOB, REVIEW THE JCL AND SPECIFY:
/*
/** LPALIB - THE DATASET NAME OF LPA LOAD LIBRARY.
/*
/** NOTE - THE COPY STEP WILL REPLACE EXISTING
/** MEMBERS OF THE SAME NAME.
/*
/** BLOAD - THE DATASET CONTAINING THE BUILDER MESSAGE MODULES
/*
//COPY EXEC PGM=IEBCOPY,REGION=2M
//SYSPRINT DD SYSOUT=*
//LPALIB DD DSN=SYS1.LPA.LIBRARY,
//          DISP=SHR
//BLOAD DD DSN=BUILDER.R140.SMPE.T.BLSYSL,
```

---

## BLXMSG#1 (cont.)

```
//          DISP=SHR
//SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSUT4    DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN     DD *
COPY INDD=( (BLLOAD,R) ),OUTDD=LPALIB
  SELECT MEMBER=MARKM00
  SELECT MEMBER=MARKM01
  SELECT MEMBER=MARKM02
  SELECT MEMBER=MARKM03
  SELECT MEMBER=MARKM04
  SELECT MEMBER=MARKM05
  SELECT MEMBER=MARKM06
  SELECT MEMBER=MARKM07
  SELECT MEMBER=MARKM08
  SELECT MEMBER=MARKM09
  SELECT MEMBER=MARKM10
  SELECT MEMBER=MARKM11
  SELECT MEMBER=MARKM12
  SELECT MEMBER=MARKM13
  SELECT MEMBER=MARKM14
  SELECT MEMBER=MARKM15
  SELECT MEMBER=MARKM16
  SELECT MEMBER=MARKM17
  SELECT MEMBER=MARKM18
  SELECT MEMBER=MARKM19
  SELECT MEMBER=MARKM20
  SELECT MEMBER=MARKM21
  SELECT MEMBER=MARKM22
  SELECT MEMBER=MARKM23
  SELECT MEMBER=MARKM24
  SELECT MEMBER=MARKM25
  SELECT MEMBER=MARKM26
  SELECT MEMBER=MARKM27
  SELECT MEMBER=MARKM28
  SELECT MEMBER=MARKM29
  SELECT MEMBER=MARKM30
  SELECT MEMBER=MARKM31
  SELECT MEMBER=MARKM32
  SELECT MEMBER=MARKM33
  SELECT MEMBER=MARKM34
  SELECT MEMBER=MARKM35
  SELECT MEMBER=MARKM36
  SELECT MEMBER=MARKM37
  SELECT MEMBER=MARKDX
  SELECT MEMBER=MARKMIC
/*
//
```

## BLXOLX#1

```
//BLXOLX#1 JOB (ACCT)
//*
/* DEFAULT JCL
/*
/* BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/*
/******
/* BLXOLX#1 - COPY THE "ONLINE EXECUTIVE" HELP MEMBERS
/*           - TO A TSO HELP DATA SET
/******
/*
/* THIS JOB STREAM COPIES THE ONLINE EXECUTIVE (OLX, OFI, OQL)
/* TSO HELP MEMBERS TO A TSO ONLINE HELP DATASET.
/*
/* A HELP DATASET IS ALLOCATED AND THE HELP MEMBERS ARE COPIED.
/*
/* BEFORE YOU RUN THIS JOB, REVIEW THE JCL AND SPECIFY:
```

---

## BLXOLX#1 (cont.)

```
/* OUT      - THE DSN FOR THE TSO HELP DATASET.
/*
/*          IF YOU USE AN EXISTING DATASET, SKIP THE
/*          ALLOC STEP.
/*
/*          NOTE - THE COPY STEP WILL REPLACE EXISTING
/*          MEMBERS OF THE SAME NAME.
/*
/*          NOTE - THE OUT DATASET IS ALLOCATED TO A SYSDA UNIT
/*
/* BLHELP    - THE DATASET CONTAINING THE BUILDER HELP MEMBERS.
/*
/*
/*ALLOC      EXEC PGM=IEFBR14,REGION=256K
/*OUT         DD DSN=BUILDER.R140.OLXHELP,
/*            DISP=(NEW,CATLG),
/*            UNIT=SYSDA,SPACE=(TRK,(5,2,5)),
/*            DCB=(RECFM=FB,LRECL=80,BLKSIZE=8800)
/*
/*COPY        EXEC PGM=IEBCOPY,REGION=2M
/*SYSPRINT    DD SYSOUT=*
/*OUT         DD DSN=BUILDER.R140.OLXHELP,
/*            DISP=SHR
/*BLHELPS     DD DSN=BUILDER.R140.SMPE.T.BLSAMP,
/*            DISP=SHR
/*SYSUT3      DD UNIT=SYSDA,SPACE=(CYL,1)
/*SYSUT4      DD UNIT=SYSDA,SPACE=(CYL,1)
/*SYSIN       DD *
/*            COPY INDD=( (BLHELPS,R) ),OUTDD=OUT
/*            SELECT MEMBER=EDITIV
/*            SELECT MEMBER=END
/*            SELECT MEMBER=LIB
/*            SELECT MEMBER=M4EXEC
/*            SELECT MEMBER=OQL
/*            SELECT MEMBER=QUERYIV
/*            SELECT MEMBER=QUIT
/*            SELECT MEMBER=RUNIV
/*            SELECT MEMBER=SUBIV
/*
/*
```

## BLXOLX#2

```
/*BLXOLX#2 JOB (ACCT)
/*
/** DEFAULT JCL
/*
/** BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/**
/******
/** BLXOLX#2 - COPY SOME OF THE "ONLINE EXECUTIVE" COMMAND
/**           - PROCESSING MODULES TO THE "SYS1.LINKLIB"
/******
/*
/** THIS JOB COPIES SOME ONLINE EXECUTIVE (OLX, OFI, OQL)
/** COMMAND PROCESSING MODULES TO THE "SYS1.LINKLIB".
/*
/** BEFORE YOU RUN THIS JOB, REVIEW THE JCL AND SPECIFY:
/*
/** LINKLIB - SYS1.LINKLIB. YOU CAN SPECIFY A DIFFERENT
/**           LIBRARY INSTEAD OF SYS1.LINKLIB.
/*
/**          NOTE - THE COPY STEP WILL REPLACE EXISTING
/**          MEMBERS OF THE SAME NAME.
/*
/** BLOAD    - THE DATASET CONTAINING THE BUILDER COMMAND
/**           PROCESSING MODULES.
/*
```

---

## BLXOLX#2 (cont.)

```
//COPY      EXEC  PGM=IEBCOPY,REGION=2M
//SYSPRINT  DD   SYSOUT=*
//LINKLIB   DD   DSN=SYS1.LINKLIB,DISP=SHR
//          DD   DISP=SHR
//BLOAD     DD   DSN=BUILDER.R140.SMPE.T.BLSYSL,
//          DD   DISP=SHR
//SYSUT3    DD   UNIT=SYSDA,SPACE=(CYL,1)
//SYSUT4    DD   UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN     DD   *
COPY INDD=( (BLOAD,R) ),OUTDD=LINKLIB
      SELECT MEMBER=M4EXEC
      SELECT MEMBER=M4EXECCE
      SELECT MEMBER=M4EXECCI
      SELECT MEMBER=M4EXECCP
      SELECT MEMBER=M4EXECCCT
      SELECT MEMBER=M4EXECSE
      SELECT MEMBER=M4EXECVT
      SELECT MEMBER=M4EXECXR
/*
//
```

## BLXPAL#1

```
//BLXPAL#1 JOB (ACCT)
//*
//*  DEFAULT JCL
//*
//*  BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXPAL#1 - CATALOG PAL (PROGRAM ANALYZER) DEFINITIONS AND
//*             - PROCESSING REQUESTS INTO A "M4LIB"
//*****
//*
//*  THIS JOB STREAM ALLOCATES AN M4LIB AND CATALOGS THE
//*  PROGRAM ANALYZER (PAL) FILE DEFINITIONS AND APPLICATION REQUESTS.
//*
//*  BEFORE YOU RUN THIS JOB, REVIEW THE JCL AND SPECIFY:
//*
//*  JOBLIB - THE BUILDER AND COMLIB LOAD LIBRARIES.
//*
//*  M4LIB - THE DSN FOR THE LIBRARY THAT WILL CONTAIN THE
//*          CATALOGED PAL FILE DEFINITIONS AND REQUESTS.
//*
//*          THIS IS A BDAM TYPE LIBRARY WITH MINIMUM SPACE
//*          ALLOCATED ON UNIT SYSDA.
//*
//*  M4INPUT - THE DATASET/MEMBER THAT CONTAINS THE PAL FILE
//*             DEFINITIONS AND REQUESTS.
//*
//JOBLIB     DD   DSN=BUILDER.R140.SMPE.T.BLSYSL
//*
//INIT      EXEC  PGM=MARKINIT,REGION=1M
//M4LIST     DD   SYSOUT=*
//M4LIB      DD   DSN=BUILDER.R140.PAL.M4LIB,
//          DD   DISP=(NEW,CATLG),
//          DD   SPACE=(TRK,2,,CONTIG),UNIT=SYSDA
//*
//DEFRUN     EXEC  PGM=MARKIV,REGION=2M
//M4LIB      DD   DSN=BUILDER.R140.PAL.M4LIB,
//          DD   DISP=SHR
//M4LIST     DD   SYSOUT=*
//M4INPUT    DD   DSN=BUILDER.R140.SMPE.T.BLSAMP(PALFDS),
//          DD   DISP=SHR
//*
//PROCRUN    EXEC  PGM=MARKIV,REGION=2M
```

---

## BLXPAL#1 (cont.)

```
//M4LIST      DD SYSOUT=*
//M4LIB       DD DSN=BUILDER.R140.PAL.M4LIB,
//            DISP=SHR
//M4INPUT     DD DSN=BUILDER.R140.SMPE.T.BLSAMP (PALREQS),
//            DISP=SHR
//
```

## BLXRLK#1

```
//BLXRLK#1 JOB (ACCT)
//*
/* DEFAULT JCL
/*
/* BUILT BY THE INSTALLATION PREPARATION DIALOG
/*
/*
/******
/* BLXRLK#1 - RELINK THE VISION:BUILDER OVERLAY MODULE
/*           - "MARKIV" WITH A USER "M4OWN" MODULE
/*           - FOR STATIC OWN CODE INTEGRATION
/******
/*
/* THE FOLLOWING SAMPLE JOB RELINKS THE VISION:BUILDER LOAD LIBRARY
/* MODULE NAMED "MARKIV". THIS MODULE USES THE OVERLAY FACILITY.
/*
/* THE PRIMARY PURPOSE FOR RUNNING THIS JOB IS TO ACCOMMODATE
/* STATIC OWN CODE INTEGRATION.
/*
/* THE JOB CAN BE USED WHEN:
/*   A NEW IMS INTERFACE MODULE (DFSLI000) IS NEEDED, OR
/*   A DIFFERENT LOAD LIBRARY BLKSIZE IS DESIRED.
/*
/* BEFORE RUNNING THIS JOB,
/* REVIEW THE NAMED "JCL" STATEMENTS AND SPECIFY:
/*
/*   SYSLMOD - THE LOAD LIBRARY TO CONTAIN THE RELINKED
/*             MODULE "MARKIV".
/*   LOADLIB - THE LOAD LIBRARY THAT CONTAINS THE LOAD
/*             MODULE "MARKIV".
/*   OBJLIB  - AN OBJECT OR LOAD LIBRARY THAT CONTAINS THE
/*             USER'S "M4OWN" MODULE FOR STATIC INTEGRATION.
/*
/*   NOTE: IF STATIC INTEGRATION IS NOT IMPLEMENTED,
/*         COMMENT THE "OBJLIB" DD STATEMENT. A CONDITION CODE
/*         OF 8 WILL OCCUR WHEN THE DD STATEMENT IS NOT PRESENT.
/*
/*   USE THE "BUILDER.R140.SMPE.T.BLSAMP" DATASET WHICH HAS
/*   A DEFAULT "M4OWN" MODULE TO TURN OFF STATIC OWN CODE.
/*
/*   DLILIB - THE IMS LIBRARY THAT CONTAINS THE INTERFACE
/*            MODULE "DFSLI000".
/*
/*   NOTE: IF IMS IS NOT USED AT YOUR INSTALLATION,
/*         COMMENT THE "DLILIB" DD STATEMENT. A CONDITION CODE
/*         OF 8 WILL OCCUR WHEN THE DD STATEMENT IS NOT PRESENT.
/*
/*   SYSLIN - THE "BUILDER.R140.SMPE.T.BLSAMP" DATA SET CONTAINS THE
/*            LINK EDIT CONTROL STATEMENT MEMBER.
/*
/* THIS JOB NORMALLY COMPLETES WITH A CONDITION CODE 4.
/*
/******
/* RELINK THE BUILDER MODULE "MARKIV"
/******
/*
/*RELINK EXEC PGM=HEWL,REGION=2M,
//          PARM='LET,LIST,MAP,NCAL,OVLY,XCAL'
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=BUILDER.R140.SMPE.T.BLSYSL,
```

---

## BLXRLK#1 (cont.)

```
//          DISP=SHR
//LOADLIB DD DSN=BUILDER.R140.SMPE.T.BLSYSL,
//          DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//OBJLIB DD DSN=BUILDER.R140.SMPE.T.BLSAMP,
//          DISP=SHR
//DLILIB DD DSN=IMS.RESLIB,
//          DISP=SHR
//SYSLIN DD DSN=BUILDER.R140.SMPE.T.BLSAMP(BLOVRLK),
//          DISP=SHR
//
```

## BLXRSQ#1

```
//BLXRSQ#1 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXRSQ#1 - LINK THE RESULTS QUICK START UTILITY
//*           - WITH THE CA-LIBRARIAN INTERFACE MODULES
//*****
//*
//* LINK LIBRARIAN INTERFACE MODULES WITH RESULTS QUICK START.
//*
//LBLNK PROC BLOAD=,
//          LIBLOAD=
//LINK EXEC PGM=HEWL,REGION=2M,PARM='LIST,MAP,LET,NCAL'
//SYSLIB DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//LIBSYS DD DSN=&LIBLOAD,
//          DISP=SHR
//SYSLMOD DD DSN=&BLOAD,
//          DISP=SHR
//          PEND
//*
//*****
//* BEFORE SUBMITTING THIS JCL, YOU MUST SPECIFY THE FOLLOWING
//* INFORMATION:
//*
//* BLOAD - NAME OF THE VISION:BUILDER SYSTEM LOADLIB
//* LIBLOAD - NAME OF YOUR LIBRARIAN SYSTEM LOADLIB
//*
//* NOTE: A CONDITION CODE OF 4 FROM THE LINK EDIT RUN IS OKAY.
//*
//*****
//*
//LIBLINK EXEC LBLNK,
//          BLOAD='BUILDER.R140.SMPE.T.BLSYSL',
//          LIBLOAD='LIBRARIAN.SYSTEM.LOADLIB'
//LINK.SYSLIN DD *
//          INCLUDE LIBSYS (FAIRCLS)
//          INCLUDE LIBSYS (FAIROPN)
//          INCLUDE LIBSYS (FAIRREC)
//          INCLUDE LIBSYS (FAIRMOD)
//          INCLUDE LIBSYS (FAIRERR)
//          INCLUDE LIBSYS (FAIRLOC)
//          INCLUDE LIBSYS (FAIRNTE)
//          INCLUDE LIBSYS (FAIRPNT)
//          INCLUDE LIBSYS (FAIRSEC)
//          INCLUDE SYSLMOD (DYL280LX)
//          ENTRY DYL280L
//          NAME DYL280L (R)
//*
//
```

---

## BLXRSQ#2

```
//BLXRSQ#2 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXRSQ#2 - LINK THE RESULTS QUICK START UTILITY
//*           - WITH CA-PANVALET INTERFACE MODULES
//*****
//*
//* LINK PANVALET INTERFACE MODULES WITH RESULTS QUICK START.
//*
//PNLNK  PROC BLOAD=,
//        PANLOAD=
//LINK   EXEC PGM=HEWL,REGION=2M, PARM='LIST,MAP,LET,NCAL'
//SYSLIB DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//LIBSYS DD DSN=&PANLOAD,
//        DISP=SHR
//SYSLMOD DD DSN=&BLOAD,
//        DISP=SHR
//        PEND
//*
//*****
//* BEFORE SUBMITTING THIS JCL, YOU MUST SPECIFY THE FOLLOWING
//* INFORMATION:
//*
//*   BLOAD - NAME OF VISION:BUILDER LOADLIB
//*   PANLOAD - NAME OF YOUR PANVALET SYSTEM LOADLIB
//*****
//*
//PANLINK EXEC PNLNK,
//           BLOAD='BUILDER.R140.SMPE.T.BLSYSL',
//           PANLOAD='PANVALET.SYSTEM.LOADLIB'
//LINK.SYSLIN DD *
//          INCLUDE LIBSYS(PAM)
//          INCLUDE SYSLMOD(DYL280PX)
//          ENTRY DYL280P
//          NAME DYL280P(R)
//*
//
```

## BLXRSQ#3

```
//BLXRSQ#3 JOB (ACCT)
//*
//* DEFAULT JCL
//*
//* BUILT BY THE INSTALLATION PREPARATION DIALOG
//*
//*
//*****
//* BLXRSQ#3 - RUN THE RESULTS QUICK START UTILITY
//*****
//*
//* EXECUTE THE RESULTS QUICK START UTILITY
//* ***** NOTE *****
//* THE SYSCOPY DD STATEMENT IS USED FOR MVS COPYBOOK LIBRARIES.
//* THE PANDD1 DD STATEMENT IS USED FOR PANVALET COPYBOOK LIBRARIES.
//* THE MASTER DD STATEMENT IS USED FOR LIBRARIAN COPYBOOK LIBRARIES
//*
//*****
//*
//RESLTQS PROC RGN=2M,
```

### BLXRSQ#3 (cont.)

```
//          BLOAD=,
//          DEFLIB=,
//          MEMBER=,
//          RSLTLIB=,
//          RSLTDEF=
//*
//CONVRT EXEC PGM=RESULTQS,REGION=&RGN
//STEPLIB DD DSN=&BLOAD,
//          DISP=SHR
//SYSPRINT DD SYSOUT=*
//*SYSCOPY DD DISP=SHR,DSN=USER.RESULTS.COPYLIB
//*PANDD1 DD DISP=SHR,DSN=USER.PANVALET.LIBRARY
//*MASTER DD DISP=SHR,DSN=USER.LIBR.MASTER
//SYS004 DD DISP=OLD,DSN=&DEFLIB(&MEMBER)
//SYSIN DD DISP=SHR,DSN=&RSLTLIB(&RSLTDEF)
//          PEND
//
//*****
//** FOLLOWING IS A SAMPLE EXECUTION OF THIS PROCEDURE. BEFORE YOU
//** RUN THIS PROCEDURE, SPECIFY:
//**
//** RGN - THE REGION SIZE. DEFAULT IS 2M.
//** BLOAD - THE NAME OF THE VISION:BUILDER LOADLIB
//** DEFLIB - THE LIBRARY(PDS) TO CONTAIN THE BUILDER DEFINITIONS.
//** MEMBER - THE PDS MEMBER NAME FOR THE CONVERTED VISION:BUILDER
//**          FILE DEFINITION IN THE DEFINITION LIBRARY.
//** RSLTLIB - THE PDS CONTAINING THE VISION:RESULTS FILE
//**          DEFINITION SOURCE STATEMENTS.
//** RSLTDEF - THE PDS MEMBER NAME OF THE INPUT VISION:RESULTS
//**          FILE DEFINITION TO BE CONVERTED.
//**
//** *** N O T E ***
//**
//** THIS PROCEDURE ASSUMES INPUT FROM A PDS MEMBER. OPTIONALLY, IT
//** MAY ALSO COME FROM A RESULTS COPY (MVS PDS), COPYP (PANVALET),
//** OR COPYL (LIBRARIAN) STATEMENT. IF SO, YOU MUST UN-COMMENT THE
//** APPROPRIATE SYSCOPY (MVS PDS), PANDD1 (PANVALET), OR MASTER
//** (LIBRARIAN) DD STATEMENT IN THE PROCEDURE, SPECIFYING THE
//** PROPER DATA SET NAME FOR THE LIBRARY USED. PLEASE REFER TO THE
//** MANUAL FOR DETAILS IN SETTING UP COPY SUPPORT.
//**
//*****
//STEP01 EXEC RESLTQS,RGN=2M,
//          BLOAD='BUILDER.R140.SMPE.T.BLSYSL',
//          DEFLIB='YOUR.DEFINITION.OUT.LIB',
//          MEMBER=FILENAME,
//          RSLTLIB='VISION.RESULTS.FILEDEFS',
//          RSLTDEF=FILENAME
//
```

# VISION:Builder Parameter Modules

---

VISION:Builder provides default values for many of the parameters. These values are usually determined by the operating environment in each installation. These parameters affect various functions of the system. Because these parameters are part of the installation process, VISION:Builder provides you with the capability of changing their default values.

This appendix describes the following modules:

- [M4PARAMS and M4LEPARM on page B-1](#)
- [M4SFPARM on page B-17](#)
- [MARKLIBP on page B-21](#)
- [MARKSQL on page B-23](#)
- [Query Language Parameters – BQLPARM on page B-30](#)
- [Online Language Parameters – OQLPARM on page B-37](#)

## M4PARAMS and M4LEPARM

M4PARAMS and M4LEPARM are used for changing default values. Both are supplied with VISION:Builder as Assembly Language source CSECTs and are well documented in their source form. A list of the CSECTs follows; the default is clearly indicated for each parameter.

You can replace any of the parameters as wanted, but do not make any changes that would modify the relative location of any field. The CSECT is link edited as a load module after the installation of VISION:Builder. It can be changed at any time and the last version included in the system determines the installation standards. Multiple versions of M4PARAMS or M4LEPARM can be maintained in separate partitioned data sets. If this is done, the JOBLIB statement for M4PARAMS or M4LEPARM must precede the one for VISION:Builder (concatenation) when VISION:Builder is executed. This ensures that the alternate version is used.

M4PARAMS and M4LEPARM are supplied with each new release of VISION:Builder and, as improvements and extensions are included in the system, they may change to reflect these modifications. Therefore, it is necessary to link edit the new version of M4PARAMS or M4LEPARM with each new release. Any changes to M4PARAMS or M4LEPARM are explained in the [VISION:Builder for OS/390 Getting Started Guide](#) that accompanies each new release of VISION:Builder.

Refer any questions concerning M4PARAMS or M4LEPARM to Computer Associates Technical Support. See [Contacting Computer Associates on page 1-11](#) for more information.

**Note:** The special symbols in this M4PARAMS table are for the PN print chain. Installations not using a PN chain must change the symbols accordingly.

Parameter	Parameter Name	VISION:Builder Standard
User ID	USERID	Installation identification from Computer Associates.
Delimiter	DELIMITR	#
Page height	HEIGHT	66 lines
M4LIST width – Default width of page	LSTWIDTH	132 print positions
Default width of page	LSTDFWOP	0 (M4LIST width)
Automatic GRAND summaries are printed on a separate page at the end of each report	AUTOGRND	Automatic GRAND summaries are not generated.
Report column heading border character	HEADCHAR	- (hyphen)
Repeating (S-type) subtitles	SUBTITLE	S-type subtitles do not repeat on page overflow.
Symbol for invalid field due to computation	INVALID	*
Symbol for non-existent field	NOTEXIST	- (hyphen)
Symbol for field that cannot be edited	NOTEDIT	+
Percent sign	PERCENT	%

Parameter	Parameter Name	VISION:Builder Standard
Summary labels <b>Note:</b> Each Summary and Page label can contain up to five characters.	TOTAL, CUM, COUNT, MAX, MIN, AVG, RATIO, PCT, PAGE, GRAND	TOTAL, CUM., COUNT, MAX., MIN., AVG., RATIO, PCT., PAGE, GRAND
Left formatting delimiter for source listing	LEFTMRK	(
Right formatting delimiter for source listing	RIGHTMRK	)
Double delimiter (used when both left and right formatting delimiter fall in the same place)	SINGSEP	,
Source card listing control	SLCCTL	ASA control character blank (single space)
Message control	PRINT,CONSOLE	Messages print on printer. Occasional messages print on console but only if operator action is required.
Block size for M4REPO	REPOSIZ	4096  The default value for the subfile blocking factor for variable blocked records is:  [M4REPO block size]  The default value for the subfile blocking factor for undefined blocked records is:  [M4REPO block size] – 8
Number of buffers for input files	INPUT	2
Number of buffers for output files	OUTPUT	1
One-step report storage	REPTSIZE	8192 (8192 KB)– The amount of storage allocated to the report phase of a single-step no-sort processing run.

Parameter	Parameter Name	VISION:Builder Standard
One-step sort storage	SORTSIZE	524288
Characters in edit patterns:		
■ Digit select character	DIGCHAR	9
■ Zero suppress character	ZSPCHAR	Z
■ Currency symbol character	CURCHAR	\$
■ Plus symbol character	PLUCHAR	+
■ Minus symbol character	MINCHAR	–
■ Check protection character	CKPCHAR	*
■ Decimal point character	DECCHAR	.
■ Grouping character	GRPCHAR	,
GRAPHICS:		
■ Primary plot character	PRMCHAR	X
■ Secondary plot character	SCDCHAR	*
■ Fit plot character	FITCHAR	.
■ Horizontal axis character	HZACHAR	_ (underscore)
■ Horizontal hash character	HZHCHAR	(vertical bar)
■ Vertical axis character	VTACHAR	(vertical bar)
■ Vertical hash character	VTHCHAR	- (hyphen)
Time Processing in models 4260, 4360, and 4460:		
■ First conversion factor	MULTPLR1	60 – indicating minutes per hour.
■ Second conversion factor	MULTPLR2	60 – indicating seconds per min.
■ Units delimiter	TIMEDELM	: (colon)
Months	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC

Parameter	Parameter Name	VISION:Builder Standard
DATE flag format	DATE	MMM DD, YYYY
TODAY flag format	M4TODAY	MMDDYY
DATE flag delimiters:		
■ TODAY	TODAYDLM	/ (that is, MM/DD/YY)
■ ISDATE	ISDATDLM	- (that is, YYYY-MM-DD)
■ JULIAN	JULDLM	. (that is, YY.DDD)
Sort program	SORTPGM	5740-SM1
Maximum working storage	MAXGETMN	1024 (1024 KB)
Minimum main storage released using FREEMAIN	MINCORE	12 (12 KB)
GRAPHICS:		
■ Alternate M4LIST width	ALTWIDTH	132 print positions
■ Alternate default width of page	ALTDFWOP <b>Note:</b> A value of zero (0) in ALTDFWOP causes the system to use the alternate M4LIST width for this specification.	0 (M4LIST1 width)
Suppress no-data-selected report	SUPRSNDS	N – The skeleton report is printed.
Default codes to print the information and warning messages for run phases	DECMSOPT	Y – Yes
	PROMSOPT	Y – Yes
	RPTMSOPT	Y – Yes
File processing AMODE(31)	AMODE31	Y – Yes
M4PAOUT maximum lines	PALTRCMX	1024 lines of space
High level ISAM index control	COREINDX	0 (no indices in storage)

Parameter	Parameter Name	VISION:Builder Standard
Default condition codes	CONDCOD1	0 – Normal
	CONDCOD2	4 – Error termination
	CONDCOD3	8 – No sorting (specified)
	CONDCOD4	16 – No sorting (invalid requests)

### M4PARAMS Source Code

```

MPOVS    TITLE 'M4PARAMS - COMPUTER ASSOCIATES INTERNATIONAL, INC.'    00010000
          ISEQ 73,80                                                    00020000
*****
*                                                                 * 00030000
*                                                                 * 00040000
*          PROPRIETARY AND CONFIDENTIAL INFORMATION OF          * 00050000
*          COMPUTER ASSOCIATES INTERNATIONAL, INC.              * 00060000
*          USE RESTRICTED BY WRITTEN LICENSE AGREEMENT           * 00070000
*                                                                 * 00080000
*          DO NOT REMOVE THIS NOTICE                            * 00090000
*                                                                 * 00100000
*                                                                 * 00110000
*          COPYRIGHT (C) COMPUTER ASSOCIATES INTERNATIONAL, INC. * 00120000
*          AS AN UNPUBLISHED WORK.  ALL RIGHTS RESERVED.         * 00130000
*                                                                 * 00140000
*****                                                                 00150000
          SPACE 3                                                    00160000
          MACRO                                                    C265 00170000
          M4TODAY &FORMAT                                           C265 00180000
          LCLA  &ACCUM                                              C265 00190000
&ACCUM   SETA  0                                                    C265 00200000
          AIF  ('&FORMAT' EQ 'MDDYY').EQU                          C265 00210000
&ACCUM   SETA  4                                                    C265 00220000
          AIF  ('&FORMAT' EQ 'DDMMYY').EQU                         C265 00230000
&ACCUM   SETA  8                                                    C265 00240000
          AIF  ('&FORMAT' EQ 'YYMMDD').EQU                         C265 00250000
&ACCUM   SETA  12                                                   C265 00260000
          AIF  ('&FORMAT' EQ 'YYDDMM').EQU                         C265 00270000
&ACCUM   SETA  16                                                   C265 00280000
          AIF  ('&FORMAT' EQ 'DDYYMM').EQU                         C265 00290000
&ACCUM   SETA  20                                                   C265 00300000
          AIF  ('&FORMAT' EQ 'MMYYDD').EQU                         C265 00310000
          MNOTE 8, 'ILLEGAL TODAY FORMAT, MDDYY ASSUMED'          C265 00320000
&ACCUM   SETA  0                                                    C265 00330000
          .EQU  ANOP                                                C265 00340000
TODAY    EQU  &ACCUM                                                C265 00350000
          MEND                                                    C265 00360000
          EJECT                                                    00370000
*****                                                                 00380000
*          USERS MUST NOT MAKE CHANGES BEFORE THIS PAGE        C265 00390000
*****                                                                 00400000
          EJECT                                                    00410000
M4PARAMS CSECT                                                    00420000
*****                                                                 00430000
*                                                                 * 00440000
*          THIS ROUTINE CONTAINS ALL PARAMETERS WHICH MAY BE SET AS USER * 00450000
*          OPTIONS.                                              * 00460000
*          1. USERS MAY CHANGE ANY OF THE ITEMS WITHIN THE RANGES SPECIFIED. * 00470000
*          2. USERS MUST NOT CHANGE THE LENGTH OF ANY ASSEMBLY ITEMS.      * 00480000
*          3. USERS MUST NOT CHANGE ITEMS THAT PRECEDE THE PAGE MARKED      * 00490000
*          " * USERS MUST NOT MAKE CHANGES BEFORE THIS PAGE * ."          * 00500000
*          4. USERS MUST NOT CHANGE ITEMS THAT FOLLOW THE PAGE MARKED        * 00510000
*          " * USERS MUST NOT MAKE CHANGES FOLLOWING THIS PAGE * ."        * 00520000

```

## M4PARAMS Source Code (cont.)

```

*
* THIS ROUTINE MAY BE ASSEMBLED AND LINK EDITED AFTER BUILDER
* INSTALLATION IS COMPLETE. IF ALL OF THE DEFAULT PARAMETERS
* ARE SATISFACTORY, NO ACTION IS NEEDED. OTHERWISE, THE MODIFIED
* MODULE MUST BE ASSEMBLED AND LINK EDITED ACCORDING TO THE
* INSTRUCTIONS PROVIDED IN THE INSTALLATION MANUAL.
*
*****
EJECT
*
* USER ID - THIRTY-TWO CHARACTERS OF TEXT TO PRINT IN THE
* SIGN ON.
*
USERID DC CL32' '
SPACE 5
*
* SYSTEM - ANY VALID PRINTABLE OR UNPRINTABLE CHARACTER 11.0
* DELIMITER EXCEPT UNDERSCORE (X'6D') AND TILDE (X'A1'). 11.0
* THIS CHARACTER IS RESERVED AND MAY NOT APPEAR 11.0
* IN ANY STATEMENTS EXCEPT FOR ITS USE
* AS A DELIMITER.
*
DELIMITR EQU C'#'
SPACE 5
DEFAULT = POUND (NUMBER) SIGN
*
* PAGE - THE NUMBER OF PRINTABLE LINES ON A PAGE. THIS
* HEIGHT NUMBER MUST BE GREATER THAN ZERO AND MUST BE
* COMPATIBLE WITH THE DEFAULT PRINTER FORM AND
* SIZE SPECIFICATION FOR THE PRINTERS AT YOUR SITE.
* THE VALUE ASSUMES A SETTING OF 6 LINES PER INCH.
*
HEIGHT EQU 66
SPACE 5
DEFAULT = 11 INCH PAGE AT 6 LPI
*
* M4LIST - THE NUMBER OF PRINTABLE COLUMNS ON THE M4LIST QN10
* WIDTH OUTPUT DEVICE, NOT INCLUDING THE ASA CONTROL QN10
* CHARACTER. THIS IS THE M4LIST RECORD LENGTH-1, QN10
* AND MUST BE AT LEAST 132 COLUMNS. QN10
*
LSTWIDTH EQU 132
DEFAULT = 132 COLUMNS
*
* DEFAULT - THE NUMBER OF PRINTABLE COLUMNS ON AN OUTPUT QN10
* WIDTH OF REPORT PAGE, NOT INCLUDING THE ASA CONTROL QN10
* PAGE CHARACTER. THIS IS THE DEFAULT VALUE USED IF QN10
* "WIDTH OF PAGE" ON THE EN/ER STATEMENT IS LEFT QN10
* BLANK. THIS VALUE MUST NOT EXCEED THE M4LIST QN10
* WIDTH (LSTWIDTH) SPECIFIED ABOVE. QN10
* NOTE: A ZERO VALUE WILL CAUSE THE SYSTEM TO QN10
* USE M4LIST WIDTH FOR THIS SPECIFICATION. QN10
*
LSTDFWOP EQU 0
EJECT
DEFAULT = M4LIST WIDTH
*
* AUTOMATIC - THE AUTOMATIC GRAND SUMMARIES FEATURE PROVIDES
* GRAND GRAND SUMMARIES FOR ALL FIELDS FOR WHICH A
* SUMMARIES SUMMARY HAS BEEN REQUESTED ON A REPORT. THE
* FEATURE IS ACTIVATED BY ENTERING AN 8. THE
* FEATURE IS DEACTIVATED BY ENTERING A 0.
*
AUTOGRND EQU 0
SPACE 5
DEFAULT - NO AUTO GRAND SUMS
*
* REPORT - ANY VALID PRINTABLE OR UNPRINTABLE CHARACTER.
* COLUMN THIS CHARACTER WILL BE USED TO FORM THE LINES
* HEADING AROUND COLUMNS HEADINGS ON REPORTS. IF A BLANK
* CHARACTER IS SPECIFIED, ONE BLANK LINE WILL BE PRINTED
* BETWEEN THE COLUMN HEADINGS AND THE DETAIL LINES
* FOR SINGLE-SPACED REPORTS, TWO FOR DOUBLE-SPACED
* REPORTS, ETC.
*
HEADCHAR EQU C'-'
DEFAULT = DASH (HYPHEN)

```

## M4PARAMS Source Code (cont.)

```

SPACE 5 01240000
* 01250000
* S-TYPE - THE REPEATING SUBTITLE FEATURE PROVIDES FOR 01260000
* SUBTITLE THE PRINTING OF THE PREVIOUS S-TYPE SUBTITLE 01270000
* CONTROL UPON THE COMPLETION OF PAGE OVERFLOW. THE 01280000
* FEATURE IS ACTIVATED BY ENTERING A 1. THE 01290000
* FEATURE IS DEACTIVATED BY ENTERING A 0. 01300000
* 01310000
SUBTITLE EQU 0 DEFAULT = NO REPEATED SUBTITLES 01320000
EJECT 01330000
* 01340000
* SPECIAL - THESE CHARACTERS ARE PRINTED WHEN SPECIAL 01350000
* OUTPUT SITUATIONS OCCUR DURING REPORTING: 01360000
* CHARACTERS (1) FIELD IS INVALID 01370000
* (2) FIELD DOES NOT EXIST 01380000
* (3) FIELD CANNOT BE EDITED (EITHER WILL NOT 01390000
* CONVERT OR IS TOO BIG FOR THE COLUMN) 01400000
* 01410000
INVALID EQU C'*' DEFAULT = STAR FOR INVALID 01420000
NOTEXIST EQU C'-' DEFAULT = DASH FOR MISSING 01430000
NOTEDIT EQU C'+' DEFAULT = PLUS FOR UNEDITABLE 01440000
SPACE 5 01450000
* 01460000
* PERCENT - THIS CHARACTER IS PRINTED FOLLOWING A PERCENT 01470000
* CHARACTER SUMMARY VALUE (E.G., 75.25%). J145 01480000
* 01490000
PERCENT EQU C'%' DEFAULT = PERCENT SIGN 01500000
SPACE 5 01510000
* 01520000
* SUMMARY - THIS TABLE CONTAINS ONE FIVE-CHARACTER ENTRY 01530000
* LABEL FOR EACH TYPE OF SUMMARY, PLUS ENTRIES FOR 01540000
* TABLE PAGE AND GRAND. EACH ENTRY MUST BE EXACTLY 01550000
* FIVE CHARACTERS LONG. LEADING OR TRAILING 01560000
* BLANKS ARE ACCEPTABLE. 01570000
* 01580000
TOTAL ORG M4PARAMS+105 ***** DO NOT CHANGE THIS STATEMENT SYSM 01590000
DC CL5'TOTAL' 01600000
CUM DC CL5'CUM. ' 01610000
COUNT DC CL5'COUNT' 01620000
MAX DC CL5'MAX. ' 01630000
MIN DC CL5'MIN. ' 01640000
AVG DC CL5'AVG. ' 01650000
RATIO DC CL5'RATIO' 01660000
PCT DC CL5'PCT. ' 01670000
PAGE DC CL5'PAGE ' 01680000
GRAND DC CL5'GRAND' 01690000
EJECT 01700000
* 01710000
* LISTING - THESE CHARACTERS ARE USED AS SEPARATORS IN 01720000
* DELIMITER THE FORMATTED SOURCE STATEMENT LISTING. THE 01730000
* CHARACTERS SINGLE SEPARATOR IS USED WHEN A LEFT AND 01740000
* RIGHT SEPARATOR WOULD OTHERWISE OCCUPY THE 01750000
* SAME POSITION. 01760000
* 01770000
LEFTMRK EQU C'(' DEFAULT = LEFT PARENTHESIS 01780000
RIGHTMRK EQU C')' DEFAULT = RIGHT PARENTHESIS 01790000
SINGSEP EQU C',' DEFAULT = COMMA 01800000
SPACE 5 01810000
* 01820000
* SOURCE - ASA CARRIAGE CONTROL CHARACTER FOR FORMATTED 01830000
* STATEMENT SOURCE STATEMENT LISTING. THIS CARRIAGE 01840000
* LISTING CONTROL CHARACTER IS USED ON ALL FORMATTED 01850000
* VERTICAL SOURCE LINES AND ON THE FIRST LINE OF ANY 01860000
* SPACING FORMATTED SOURCE STATEMENT COLUMN HEADINGS. 01870000
* ACCEPTABLE CARRIAGE CONTROL CHARACTERS ARE: 01880000
* BLANK = SINGLE SPACING 01890000
* 0 = DOUBLE SPACING 01900000
* - = TRIPLE SPACING 01910000
* 01920000
SLCCTL EQU C' ' DEFAULT = SINGLE SPACING 01930000
SPACE 5 01940000

```

## M4PARAMS Source Code (cont.)

```

* 01950000
* MESSAGE - THE MESSAGE CONTROL FEATURE PROVIDES FOR 01960000
* CONTROL SUPPRESSION OF MESSAGES OUTPUT TO M4LIST 01970000
* AND/OR THE CONSOLE TYPEWRITER. MESSAGES 01980000
* ARE INHIBITED BY ENTERING A 1. MESSAGES 01990000
* ARE NOT INHIBITED BY ENTERING A 0. 02000000
* 02010000
PRINT EQU 0 DEFAULT = PRINTER MESSAGES ON 02020000
CONSOLE EQU 1 DEFAULT = CONSOLE MESSAGES OFF 02030000
EJECT 02040000
* 02050000
* M4REPO - THE BLOCKSIZE FOR REPORT FILES. THE SIZE 02060000
* BLOCKSIZE MUST BE AT LEAST 264. NOTE THAT THIS J228 02070000
* BLOCKSIZE IS ALSO USED AS THE DEFAULT 02080000
* BLOCKSIZE FOR VARIABLE LENGTH SUBFILES QN06 02090000
* AND PROGRAM ANALYZER (PAL) OUTPUT. QN06 02100000
* ALSO, THIS BLOCKSIZE - 8 IS USED AS QN06 02110000
* THE DEFAULT BLOCKSIZE FOR UNDEFINED 02120000
* LENGTH SUBFILES. 02130000
* 02140000
REPOSIZ EQU 4096 DEFAULT = 4096 BLOCKSIZE 02150000
SPACE 5 02160000
* 02170000
* NUMBER - NUMBER OF I/O BUFFERS TO BE ASSIGNED 02180000
* OF I/O TO USER DATA FILES. 02190000
* BUFFERS 02200000
* 02210000
INPUT EQU 2 DEFAULT = 2 INPUT BUFFERS/FILE 02220000
OUTPUT EQU 1 DEFAULT = 1 OUTPUT BUFFER/FILE 02230000
SPACE 3 RIP 02240000
* RIP 02250000
* ONE-STEP - THIS IS THE DEFAULT MAIN STORAGE ALLOCATION FOR RIP 02260000
* REPORT THE REPORTER WHEN REPORT FILE OPTIMIZATION RIP 02270000
* STORAGE IS USED IN A NO-SORT TYPE RUN. RIP 02280000
* RIP 02290000
* THIS VALUE MUST BE AT LEAST 1024 AND NO GREATER RIP 02300000
* THAN 1048576. RIP 02310000
REPTSIZE EQU 8192 DEFAULT = 8K RIP 02320000
SPACE 3 SIP 02330000
* SIP 02340000
* ONE-STEP - THIS IS THE DEFAULT MAIN STORAGE ALLOCATION FOR SIP 02350000
* SORT THE SORT PROGRAM WHEN REPORT FILE OPTIMIZATION SIP 02360000
* STORAGE IS USED IN A RUN REQUIRING A SORT OF THE SIP 02370000
* REPORT FILE. SIP 02380000
* THIS VALUE MUST BE AT LEAST 1024 AND NO GREATER SIP 02390000
* THAN 16777216. SIP 02400000
* SIP 02410000
SORTSIZE EQU 524288 DEFAULT = 512K SIP 02420000
EJECT SIP 02430000
***** ULS 02440000
* ULS 02450000
* C H A R A C T E R S U S E D I N E D I T P A T T E R N S * U026 02460000
* ULS 02470000
* THE FOLLOWING EIGHT M4PARAMS OPTIONS CONTROL THE FORMAT OF * ULS 02480000
* FIELDS THAT ARE USING EXPLICIT EDIT PATTERNS. * U026 02490000
* ULS 02500000
* THE EIGHT PARAMETERS ARE: DIGIT SELECT CHARACTER. DIGCHAR* ULS 02510000
* ZERO SUPPRESS CHARACTER. ZSPCHAR* ULS 02520000
* CURRENCY SYMBOL CHARACTER. CURCHAR* ULS 02530000
* PLUS SYMBOL CHARACTER. PLUCHAR* ULS 02540000
* MINUS SYMBOL CHARACTER. MINCHAR* ULS 02550000
* CHECK PROTECTION CHARACTER. CKPCHAR* ULS 02560000
* DECIMAL POINT CHARACTER. DECCHAR* ULS 02570000
* GROUPING CHARACTER. GRPCHAR* ULS 02580000
* ULS 02590000
* THE DECIMAL POINT AND GROUPING CHARACTERS ARE ALSO USED FOR * ULS 02600000
* OUTPUT REPORT EDITING OF NUMERIC FIELDS WHEN NO EXPLICIT * U026 02610000
* PATTERN IS SPECIFIED AND ARE RECOGNIZED AS DECIMAL POINT * ULS 02620000
* AND GROUPING CHARACTERS RESPECTIVELY WHEN CONVERTING INPUT * ULS 02630000
* CHARACTER STRING DATA TO NUMERIC VALUES. * ULS 02640000
* ULS 02650000

```

## M4PARAMS Source Code (cont.)

```

*
*   VALID ENTRIES FOR THESE PARAMETERS INCLUDE ANY CHARACTER
*   PRINTABLE OR UNPRINTABLE EXCEPT CHARACTERS IN THE RANGE
*   OF HEXADECIMAL VALUES X'00' THROUGH X'30'.  IN ADDITION,
*   EACH SYMBOL MUST BE UNIQUE AMONG ALL THE EIGHT EDITING
*   SYMBOLS AND THE SYSTEM DELIMITER.  THAT IS, NONE OF THE
*   NINE PARAMETERS (EIGHT EDIT SYMBOLS AND ONE SYSTEM
*   DELIMITER) MAY BE THE SAME CHARACTER.
*
*
*****
          SPACE 3
*
*   DIGIT      - SPECIFIES A DIGIT POSITION IN NUMERIC EDITED
*   SELECT      FIELDS.
*   CHARACTER
*
DIGCHAR  EQU  C'9'          DEFAULT = 9
          SPACE 3
*
*   ZERO      - SPECIFIES DIGIT POSITIONS IN NUMERIC EDITED
*   SUPPRESS    FIELDS WHICH WILL BE BLANKED IF ZERO.
*   CHARACTER
*
ZSPCHAR  EQU  C'Z'          DEFAULT = Z
          SPACE 3
*
*   CURRENCY   - SPECIFIES A LEADING/FLOATING CURRENCY SYMBOL
*   SYMBOL      FOR NUMERIC EDITED FIELDS.
*   CHARACTER
*
CURCHAR  EQU  C'$'          DEFAULT = $
          SPACE 3
*
*   PLUS      - SPECIFIES A LEADING/FLOATING/TRAILING PLUS
*   SYMBOL      SYMBOL FOR NUMERIC EDITED FIELDS.
*   CHARACTER
*
PLUCHAR  EQU  C'+'          DEFAULT = +
          SPACE 3
*
*   MINUS      - SPECIFIES A LEADING/FLOATING/TRAILING MINUS
*   SYMBOL      SYMBOL FOR NUMERIC EDITED FIELDS.
*   CHARACTER
*
MINCHAR  EQU  C'-'          DEFAULT = -
          SPACE 3
*
*   CHECK      - SPECIFIES A FILL CHARACTER FOR LEADING ZERO
*   PROTECTION  DIGITS IN NUMERIC EDITED FIELDS.
*   CHARACTER
*
CKPCHAR  EQU  C'*'          DEFAULT = *
          SPACE 3
*
*   DECIMAL    - DECIMAL POINT CHARACTER FOR NUMERIC FIELDS.
*   POINT
*   CHARACTER
*
DECCHAR  EQU  C'.'          DEFAULT = .
          SPACE 5
*
*   GROUPING   - GROUPING CHARACTER FOR NUMERIC FIELDS.
*   CHARACTER
*
GRPCHAR  EQU  C','          DEFAULT = ,
          EJECT
*****
*
*   GRAPH REPORT GRAPHING CHARACTERS

```

\* ULS 02660000  
 \* ULS 02670000  
 \* ULS 02680000  
 \* ULS 02690000  
 \* ULS 02700000  
 \* ULS 02710000  
 \* ULS 02720000  
 \* ULS 02730000  
 \* ULS 02740000  
 \* ULS 02750000  
 \* ULS 02760000  
 \* ULS 02770000  
 \* ULS 02780000  
 \* ULS 02790000  
 \* ULS 02800000  
 \* ULS 02810000  
 \* ULS 02820000  
 \* ULS 02830000  
 \* ULS 02840000  
 \* ULS 02850000  
 \* ULS 02860000  
 \* ULS 02870000  
 \* ULS 02880000  
 \* ULS 02890000  
 \* ULS 02900000  
 \* ULS 02910000  
 \* ULS 02920000  
 \* ULS 02930000  
 \* ULS 02940000  
 \* ULS 02950000  
 \* ULS 02960000  
 \* ULS 02970000  
 \* ULS 02980000  
 \* ULS 02990000  
 \* ULS 03000000  
 \* ULS 03010000  
 \* ULS 03020000  
 \* ULS 03030000  
 \* ULS 03040000  
 \* ULS 03050000  
 \* ULS 03060000  
 \* ULS 03070000  
 \* ULS 03080000  
 \* ULS 03090000  
 \* ULS 03100000  
 \* ULS 03110000  
 \* ULS 03120000  
 \* ULS 03130000  
 \* ULS 03140000  
 \* ULS 03150000  
 \* ULS 03160000  
 \* ULS 03170000  
 \* ULS 03180000  
 \* ULS 03190000  
 \* ULS 03200000  
 \* ULS 03210000  
 \* ULS 03220000  
 \* ULS 03230000  
 \* ULS 03240000  
 \* ULS 03250000  
 \* ULS 03260000  
 \* ULS 03270000  
 \* ULS 03280000  
 \* ULS 03290000  
 \* ULS 03300000  
 \* ULS 03310000  
 \* ULS 03320000  
 \* GRAF 03330000  
 \* GRAF 03340000  
 \* GRAF 03350000  
 \* GRAF 03360000

## M4PARAMS Source Code (cont.)

```

*
* THE FOLLOWING SEVEN M4PARAMS OPTIONS CONTROL THE CHARACTERS
* IN PLOTTING A GRAPH.
*
* THE SEVEN PARAMETERS ARE: PRIMARY PLOT CHARACTER
* SECONDARY PLOT CHARACTER
* FIT PLOT CHARACTER
* HORIZONTAL AXIS CHARACTER
* HORIZONTAL HASH CHARACTER
* VERTICAL AXIS CHARACTER
* VERTICAL HASH CHARACTER
*
* THE ONLY RESTRICTIONS APPLY TO THE PRIMARY AND SECONDARY PLOT
* CHARACTERS WHICH CANNOT BE BLANK.
*
*
*****
SPACE 3
*
* PRIMARY - SPECIFIES THE CHARACTER TO USE WHEN PLOTTING
* PLOT SINGLE POINTS (SCATTER DIAGRAM)
* CHARACTER BARS
PRMCHAR EQU C'X' DEFAULT = X
SPACE 3
*
* SECONDARY - SPECIFIES THE CHARACTER TO USE WHEN PLOTTING
* PLOT OVERLAID POINTS (SCATTER DIAGRAM)
* CHARACTER
SCDCHAR EQU C'*' DEFAULT = * (ASTERISK)
SPACE 3
*
* FIT - SPECIFIES THE CHARACTER TO USE WHEN PLOTTING
* PLOT A LEAST SQUARES FIT LINE.
* CHARACTER
FITCHAR EQU C'.' DEFAULT = . (PERIOD)
SPACE 3
*
* HORIZONTAL - SPECIFIES THE CHARACTER TO USE WHEN PLOTTING
* AXIS THE HORIZONTAL AXES.
* CHARACTER
HZACHAR EQU C'_' DEFAULT = _ (UNDERSCORE)
SPACE 3
*
* HORIZONTAL - SPECIFIES THE CHARACTER TO USE WHEN PLOTTING
* HASH THE HORIZONTAL HASH CHARACTERS MARKING
* CHARACTER INTERVALS.
HZHCHAR EQU C'|' DEFAULT = | (VERTICAL BAR)
SPACE 3
*
* VERTICAL - SPECIFIES THE CHARACTER TO USE WHEN PLOTTING
* AXIS THE VERTICAL AXES.
* CHARACTER
VTACHAR EQU C'|' DEFAULT = | (VERTICAL BAR)
SPACE 3
*
* VERTICAL - SPECIFIES THE CHARACTER TO USE WHEN PLOTTING
* HASH THE VERTICAL HASH CHARACTERS MARKING INTERVALS.
* CHARACTER
VTHCHAR EQU C'-' DEFAULT = - (DASH)
EJECT
*
* UNIT - TIME PROCESSING CAPABILITY UNIT

```

```

* GRAF 03370000
* GRAF 03380000
* GRAF 03390000
* GRAF 03400000
* GRAF 03410000
* GRAF 03420000
* GRAF 03430000
* GRAF 03440000
* GRAF 03450000
* GRAF 03460000
* GRAF 03470000
* GRAF 03480000
* GRAF 03490000
* GRAF 03500000
* GRAF 03510000
* GRAF 03520000
* GRAF 03530000
* GRAF 03540000
* GRAF 03550000
* GRAF 03560000
* GRAF 03570000
* GRAF 03580000
* GRAF 03590000
* GRAF 03600000
* GRAF 03610000
* GRAF 03620000
* GRAF 03630000
* GRAF 03640000
* GRAF 03650000
* GRAF 03660000
* GRAF 03670000
* GRAF 03680000
* GRAF 03690000
* GRAF 03700000
* GRAF 03710000
* GRAF 03720000
* GRAF 03730000
* GRAF 03740000
* GRAF 03750000
* GRAF 03760000
* GRAF 03770000
* GRAF 03780000
* GRAF 03790000
* GRAF 03800000
* GRAF 03810000
* GRAF 03820000
* GRAF 03830000
* GRAF 03840000
* GRAF 03850000
* GRAF 03860000
* GRAF 03870000
* GRAF 03880000
* GRAF 03890000
* GRAF 03900000
* GRAF 03910000
* GRAF 03920000
* GRAF 03930000
* GRAF 03940000
* GRAF 03950000
* GRAF 03960000
* GRAF 03970000
* GRAF 03980000
* GRAF 03990000
* GRAF 04000000
* GRAF 04010000
* GRAF 04020000
* GRAF 04030000
* GRAF 04040000
* GRAF 04050000
* GRAF 04060000
* GRAF 04070000

```

**M4PARAMS Source Code (cont.)**

```

* CONVERSION      CONVERSION MULTIPLIERS AND DELIMITER.      04080000
* MULTIPLIERS     THE MULTIPLIER VALUES MUST BE POSITIVE     04090000
* AND DELIMITER   INTEGERS LESS THAN 100. THE DEFAULT         04100000
*                VALUES ARE SET FOR HOURS/MINUTES/SECONDS.    04110000
*                04120000
MULTPLR1 EQU      60                      DEFAULT = 60 MINUTES/HOUR 04130000
MULTPLR2 EQU      60                      DEFAULT = 60 SECONDS/MINUTE 04140000
TIMEDELM EQU      C': '                  DEFAULT = HH:MM:SS        04150000
EJECT                                                     04160000
*                04170000
* MONTH          - THIS TABLE CONTAINS ONE THREE-CHARACTER ENTRY 04180000
* TABLE         FOR EACH MONTH OF THE YEAR. EACH ENTRY MUST BE 04190000
*                EXACTLY THREE CHARACTERS LONG. LEADING OR TRAILING 04200000
*                BLANKS ARE ACCEPTABLE.                          04210000
*                04220000
*                ORG M4PARAMS+69 ***** DO NOT CHANGE THIS STATEMENT  SYSM 04230000
JAN      DC      CL3'JAN'                  04240000
FEB      DC      CL3'FEB'                  04250000
MAR      DC      CL3'MAR'                  04260000
APR      DC      CL3'APR'                  04270000
MAY      DC      CL3'MAY'                  04280000
JUN      DC      CL3'JUN'                  04290000
JUL      DC      CL3'JUL'                  04300000
AUG      DC      CL3'AUG'                  04310000
SEP      DC      CL3'SEP'                  04320000
OCT      DC      CL3'OCT'                  04330000
NOV      DC      CL3'NOV'                  04340000
DEC      DC      CL3'DEC'                  04350000
SPACE 5                                           04360000
*                04370000
* DATE FLAG      - SPECIFIES THE FORMAT OF THE DATE FLAG. THE FORMATS 04380000
* FORMAT         AVAILABLE AND THE VALUES ENTERED TO SELECT THEM ARE: 04390000
*                MMM DD, YYYY 0          04400000
*                DD MMM YYYY 1          04410000
*                YYYY MMM DD 2          04420000
*                04430000
DATE      EQU      0                      DEFAULT = MMM DD, YYYY 04440000
SPACE 5                                           04450000
*                04460000
* TODAY FLAG     - SPECIFIES THE FORMAT OF THE TODAY FLAG. THE FORMATS 04470000
* FORMAT         AVAILABLE AND THE VALUES ENTERED TO SELECT THEM ARE: 04480000
*                MMDDYY MMDDYY          04490000
*                DDMMYY DDMMYY          04500000
*                YYMMDD YYMMDD          04510000
*                MMYYYD MMYYYD          04520000
*                DDYYMM DDYYMM          04530000
*                YYDDMM YYDDMM          04540000
*                04550000
M4TODAY MMDDYY          DEFAULT = MMDDYY          04560000
EJECT                                                     04570000
*                04580000
* TODAY FLAG     - SPECIFIES THE DELIMITER USED TO FORMAT THE TODAY 04590000
* DELIMITER      FLAG WHEN USED AS A REPORT DATE OR FREE FORM DATE 04600000
*                FLAG. ALSO USED TO FORMAT A USER-SUPPLIED REPORT 04610000
*                DATE.                          04620000
*                04630000
TODAYDLM EQU      C'/'                  DEFAULT = MM/DD/YY          04640000
SPACE 5                                           04650000
*                04660000
* ISDATE FLAG    - SPECIFIES THE DELIMITER USED TO FORMAT THE ISDATE 04670000
* DELIMITER      FLAG WHEN USED AS A REPORT DATE OR FREE FORM DATE 04680000
*                FLAG.                          04690000
*                04700000
ISDATDLM EQU      C'-'                  DEFAULT = YYYY-MM-DD          04710000
SPACE 5                                           04720000
*                04730000
* JULIAN FLAG    - SPECIFIES THE DELIMITER USED TO FORMAT THE JULIAN 04740000
* DELIMITER      FLAG WHEN USED AS A REPORT DATE OR FREE FORM DATE 04750000
*                FLAG.                          04760000
*                04770000
JULDLM EQU      C'.'                  DEFAULT = YY.DDD          04780000

```

## M4PARAMS Source Code (cont.)

```

SPACE 5 04790000
* SORT - SPECIFIES THE SORT PROGRAM FOR WHICH SORT CONTROL 04800000
* PROGRAM STATEMENTS ARE TO BE GENERATED. THE SORT PROGRAMS 04810000
* AND THE VALUES ENTERED TO SELECT THEM ARE: 04820000
* SM-023 0 I136 04830000
* 5734-SM1 1 I136 04840000
* 5740-SM1 2 04850000
* 04860000
SORTPGM EQU 2 DEFAULT = 5740-SM1 04870000
EJECT 04880000
* Z021 04890000
* MAXIMUM - SPECIFIES THE MAXIMUM AMOUNT OF STORAGE, IN K, 04900000
* WORKING TO ALLOCATE FOR WORKING STORAGE. THIS STORAGE 04910000
* STORAGE DOES NOT INCLUDE FILE BUFFERS. 04920000
* 04930000
MAXGETMN EQU 1024 DEFAULT = 1024K 04940000
* 04950000
* MINIMUM - SPECIFIES MINIMUM AMOUNT OF STORAGE, IN K, TO 04960000
* STORAGE BE RELEASED TO THE SYSTEM AT THE START OF THE 04970000
* RELEASED RUN VIA THE 'FREEMAIN' MACRO. 04980000
* TO SYSTEM 04990000
* 05000000
MINCORE EQU 12 DEFAULT = 12K 05010000
SPACE 5 05020000
* ALTERNATE - THE NUMBER OF PRINTABLE COLUMNS ON THE ALTERNATE 05030000
* M4LIST M4LIST OUTPUT DEVICE, NOT INCLUDING THE ASA 05040000
* WIDTH CONTROL CHARACTER. THIS IS THE M4LIST1 RECORD 05050000
* LENGTH-1, AND MUST BE AT LEAST 24 COLUMNS. 05060000
* 05070000
ALTWIDTH EQU 132 ALT DEFAULT = 132 COLUMNS 05080000
* 05090000
* ALTERNATE - THE NUMBER OF PRINTABLE COLUMNS ON AN ALTERNATE 05100000
* DEFAULT REPORT PAGE, NOT INCLUDING THE ASA CONTROL 05110000
* WIDTH OF CHARACTER. THIS IS THE DEFAULT VALUE USED IF 05120000
* PAGE "WIDTH OF PAGE" ON THE EN/ER STATEMENT IS LEFT 05130000
* BLANK. THIS VALUE MUST NOT EXCEED THE M4LIST1 05140000
* WIDTH (ALTWIDTH) SPECIFIED ABOVE. 05150000
* NOTE: A ZERO VALUE WILL CAUSE THE SYSTEM TO USE 05160000
* ALT M4LIST WIDTH FOR THIS SPECIFICATION. 05170000
* 05180000
ALTDWFOP EQU 0 ALT DEFAULT = M4LIST1 WIDTH 05190000
SPACE 5 SNDS 05200000
* SNDS 05210000
* SUPPRESS - WHEN NO DATA IS SELECTED FOR A REPORT A SKELETON 05220000
* NO-DATA- REPORT IS PRODUCED INDICATING NO SELECTED DATA. 05230000
* SELECTED THIS PARAMETER WILL ALLOW SUPPRESSION OF THAT 05240000
* REPORT SKELETON REPORT. ENTER 'N' TO INDICATE PRINTING 05250000
* OF THE REPORT. ENTER 'Y' TO INDICATE THAT THE 05260000
* REPORT SHOULD BE SUPPRESSED. 05270000
* SNDS 05280000
SUPRSNDS EQU C'N' DEFAULT = NO 05290000
SPACE 5 X054 05300000
* SUPPRESS - THE FOLLOWING 3 SPECIFICATIONS ALLOW INFORMATION 05310000
* INFO AND AND WARNING MESSAGES (MESSAGE TYPES 0 AND 1) TO 05320000
* WARNING BE OPTIONALLY SUPPRESSED FOR ANY OF THE DECODE/ 05330000
* MESSAGES COMPILATION, FILE PROCESSING OR REPORT GENERATION 05340000
* PHASES OF VISION:UILDER OPERATION. ENTER 'Y' 05350000
* TO ALLOW ALL INFORMATION AND WARNING MESSAGES 05360000
* TO BE PRINTED FOR THE RESPECTIVE PHASE OF 05370000
* OPERATION. ENTER 'N' TO CAUSE THE INFORMATION 05380000
* AND WARNING MESSAGES TO BE SUPPRESSED FOR THE 05390000
* RESPECTIVE PHASE OF OPERATION 05400000
* 05410000
DECMISOPT EQU C'Y' DECODE/COMPILATION PHASE INFO MESSAGES = YES 05420000
PROMSOPT EQU C'Y' FILE PROCESSING PHASE INFO MESSAGES = YES 05430000
RPTMSOPT EQU C'Y' REPORT GENERATION PHASE INFO MESSAGES = YES 05440000
SPACE 5 Z007 05450000
* FILE - THIS OPTION SPECIFIES THE DEFAULT ADDRESSING 05460000
* PROCESSING MODE TO BE USED DURING THE FILE PROCESSING 05470000
* PHASE OF THE APPLICATION. ENTER 'Y' TO INDICATE 05480000
* ADDRESS THAT 31-BIT ADDRESSING BE USED AND THAT FILE 05490000

```

## M4PARAMS Source Code (cont.)

```

* MODE          BUFFERS AND OTHER FILE PROCESSING STORAGE AREAS  Z007 05500000
*              BE ALLOCATED ABOVE THE 16-MEG STORAGE LINE.      Z007 05510000
*              ENTER 'N' TO INDICATE THAT 24-BIT ADDRESSING BE  Z007 05520000
*              USED AND THAT FILE BUFFERS AND OTHER FILE        Z007 05530000
*              FILE PROCESSING STORAGE AREAS BE ALLOCATED BELOW  Z007 05540000
*              THE 16-MEG LINE.                                   Z007 05550000
*                                                            Z007 05560000
AMODE31 EQU C'Y'      FILE PROCESSING AMODE(31) = YES          Z007 05570000
SPACE 5                                                         QN06 05580000
*                                                            QN06 05590000
* M4PAOUT      - THE MAXIMUM NUMBER OF LINES TO BE              QN06 05600000
* MAXIMUM      PROVIDED FOR THE PROGRAM ANALYZER                QN06 05610000
* LINES        REQUEST EXECUTION TRACE.                          QN06 05620000
*                                                            QN06 05630000
PALTRCMX EQU 1024      DEFAULT = 1024 LINES                    QN06 05640000
EJECT                                                         QN10 05650000
* ----- QN10 05660000
* - - - - - QN10 05670000
* - S Y S T E M   D E P E N D E N T   V A L U E S - QN10 05680000
* - - - - - QN10 05690000
* ----- QN10 05700000
SPACE 3                                                         QN10 05710000
*                                                            05720000
* HIGH LEVEL   - SPECIFIES WHETHER OR NOT THE HIGHEST LEVEL ISAM 05730000
* ISAM INDEX   INDICES FOR BISAM INPUT AND BISAM INPUT/OUTPUT    05740000
* CONTROL      FILES ARE TO RESIDE IN MAIN STORAGE FOR IMPROVED   05750000
*              EFFICIENCY. THE HIGHEST LEVEL INDICES MAY BE      05760000
*              TRACK, CYLINDER, OR (IF OPTCD=M WAS SPECIFIED     05770000
*              WHEN THE ISAM FILE WAS CREATED) MASTER INDICES.   05780000
*              THE HIGHEST LEVEL INDICES ARE MADE RESIDENT BY    05790000
*              ENTERING A 1. THE HIGHEST LEVEL INDICES REMAIN    05800000
*              NON-RESIDENT BY ENTERING A 0.                      05810000
*                                                            05820000
COREINDX EQU 0          DEFAULT = NO INDICES IN STORAGE         05830000
SPACE 3                                                         11.0 05840000
***** 05850000
*          FOLLOWING ARE THE DEFAULT CONDITION CODES * 05860000
***** 05870000
SPACE 1                                                         11.0 05880000
CONDCOD1 EQU 0          NORMAL TERMINATION                      11.0 05890000
CONDCOD2 EQU 4          ERROR TERMINATION                        11.0 05900000
CONDCOD3 EQU 8          NO SORTING (RC SPECIFICATION)           11.0 05910000
CONDCOD4 EQU 16         NO SORTING (INVALID REQUESTS)           11.0 05920000
EJECT                                                            05930000
***** 05940000
*          USERS MUST NOT MAKE CHANGES FOLLOWING THIS PAGE * 05950000
***** 05960000
EJECT                                                            05970000
ORG M4PARAMS+L'USERID , QN06 05980000
M4RELNO DC CL4'14.0'      RELEASE NUMBER Y015 05990000
M4DELIM DC AL1(DELIMITR)  SYSTEM DELIMITER 06000000
M4HEIGHT DC AL1(HEIGHT)   PAGE HEIGHT      06010000
DC X'00' ***** UNUSED ***** Z019 06020000
M4AUTOG DC AL1(AUTOGRND)  AUTOMATIC GRAND SUMMARIES 06030000
M4SING DC AL1(SINGSEP)    LISTING DELIMITER CHARACTER 06040000
M4HEADER DC AL1(HEADCHAR) REPORT COLUMN HEADING CHARACTER 06050000
M4SUBTIT DC AL1(SUBTITLE) SUBTITLE CONTROL 06060000
M4MESAND DC AL1(X'FF'-10*CONSOLE) MESSAGE CONTROL 06070000
M4MESOR DC AL1(PRINT)     MESSAGE CONTROL 06080000
M4LFTMRK DC AL1(LEFTMRK)  LISTING DELIMITER CHARACTER 06090000
M4RHTMRK DC AL1(RIGHTMRK) LISTING DELIMITER CHARACTER 06100000
M4INVFLD DC AL1(INVALID)  SPECIAL OUTPUT CHARACTER 06110000
M4NONEXT DC AL1(NOTEXIST) SPECIAL OUTPUT CHARACTER 06120000
M4NOEDIT DC AL1(NOTEDIT)  SPECIAL OUTPUT CHARACTER 06130000
M4PERCNT DC AL1(PERCENT)  PERCENT CHARACTER 06140000
DC AL1(0) ** UNUSED-OLD M4LIB RESERVE QN15 06150000
M4REPO DC Y(REPOSIZ)      M4REPO BLOCKSIZE 06160000
M4INBUT DC AL1(INPUT)     NUMBER OF I/O BUFFERS 06170000
M4OTBUF DC AL1(OUTPUT)    NUMBER OF I/O BUFFERS 06180000
M4CYLOVL DC AL1(0)        # OF TRACKS FOR ISAM CYL OFLO 06190000
M4SLCTL DC AL1(SLCCTL)    SOURCE STMTN LISTING VERT SP 06200000

```

## M4PARAMS Source Code (cont.)

M4LIST	DC	AL1 (0)	M4LIST UNIT ASSIGNMENT	06210000	
M4INPUT	DC	AL1 (0)	M4INPUT UNIT ASSIGNMENT	06220000	
M4MINCOR	DC	Y (MINCORE)	MINIMUM STORAGE RELEASE TO SYS	06230000	
M4SORTP	DC	AL1 (SORTPGM)	SORT PROGRAM	06240000	
M4DECPT	DC	AL1 (DECCHAR)	DECIMAL POINT CHARACTER	06250000	
M4COMMA	DC	AL1 (GRPCHAR)	GROUPING CHARACTER	06260000	
M4MULT1	DC	AL1 (MULTPLR1)	UNIT CONVERSION MULTIPLIER	06270000	
M4MULT2	DC	AL1 (MULTPLR2)	UNIT CONVERSION MULTIPLIER	06280000	
M4TIMDEL	DC	AL1 (TIMEDELM)	UNIT CONVERSION DELIMITER	06290000	
M4DATFMT	DC	AL1 (DATE)	DATE FLAG FORMAT	06300000	
M4MONTH	EQU	*	MONTH TABLE	06310000	
	ORG	**+3*12 ***** DO NOT CHANGE THIS STATEMENT *****		06320000	
M4LABEL	EQU	*	SUMMARY LABEL TABLE	06330000	
	ORG	**+5*10 ***** DO NOT CHANGE THIS STATEMENT *****		06340000	
M4TDYFMT	DC	AL1 (TODAY)	TODAY FLAG FORMAT	06350000	
M4TDYDLM	DC	AL1 (TODAYDLM)	TODAY FLAG DELIMITER	06360000	
M4ISDDLM	DC	AL1 (ISDATDLM)	ISDATE FLAG DELIMITER	06370000	
M4JULDLM	DC	AL1 (JULDLM)	JULIAN FLAG DELIMITER	06380000	
M4VOLCNT	DC	AL1 (0)	M4REPO VOLUME COUNT	06390000	
	DC	Y (0)	** UNUSED-OLD M4LIB BLKSIZE QN15	06400000	
M4CYLIDX	DC	AL1 (COREINDX)	HIGH LEVEL ISAM INDEX CONTROL	06410000	
M4LSTLBL	DC	AL1 (0)	M4LIST TAPE LABELS	06420000	
M4AM31	DC	AL1 (AMODE31)	31-BIT ADDRESS MODE OPTION Z007	06430000	
M4INTR	DC	AL1 (0)	IMPRECISE INTERRUPT BIG1	06440000	
M4MODNO	DC	AL1 (1)	360/370 INSTRUCTION SET BIG1	06450000	
M4FLTPTNT	DC	AL1 (1)	FLOATING POINT HARDWARE BIG1	06460000	
M49DCHAR	DC	AL1 (DIGCHAR)	DIGIT SELECT ULS	06470000	
M4ZDCHAR	DC	AL1 (ZSPCHAR)	ZERO SUPPRESS ULS	06480000	
M4CPCHAR	DC	AL1 (CKPCHAR)	CHECK PROTECTION ULS	06490000	
M4CUCHAR	DC	AL1 (CURCHAR)	CURRENCY SYMBOL ULS	06500000	
M4PLCHAR	DC	AL1 (PLUCHAR)	PLUS SIGN ULS	06510000	
M4MICCHAR	DC	AL1 (MINCHAR)	MINUS SIGN ULS	06520000	
M4REPTSZ	DC	AL4 (REPTSIZE)	REPORTER STORAGE RTP	06530000	
M4SORTSZ	DC	AL4 (SORTSIZE)	SORT STORAGE STP	06540000	
M4PRCHAR	DC	AL1 (PRMCHAR)	PRIMARY PLOT CHARACTER GRAF	06550000	
M4SCCHAR	DC	AL1 (SCDCHAR)	SECONDARY PLOT CHARACTER GRAF	06560000	
M4FTCHAR	DC	AL1 (FITCHAR)	FIT PLOT CHARACTER GRAF	06570000	
M4HACHAR	DC	AL1 (HZACHAR)	HORIZONTAL AXIS CHARACTER GRAF	06580000	
M4HHCHAR	DC	AL1 (HZHCHAR)	HORIZONTAL HASH CHARACTER GRAF	06590000	
M4VACHAR	DC	AL1 (VTACHAR)	VERTICAL AXIS CHARACTER GRAF	06600000	
M4VHCHAR	DC	AL1 (VTHCHAR)	VERTICAL HASH CHARACTER GRAF	06610000	
M4PALTRM	DC	AL4 (PALTRCMX)	MAX PAL TRACE LINES QN06	06620000	
	DC	X'00'	***** UNUSED ***** Z019	06630000	
M4AWIDTH	DC	AL2 (ALTWIDTH)	ALT M4LIST WIDTH (LRECL-1) Z019	06640000	
M4SUPNDS	DC	AL1 (SUPRSNDS)	SUPPRESS NO-DATA-SEL RPT? SNDS	06650000	
	DC	X'00'	***** UNUSED ***** 11.0	06660000	
M4CCODE1	DC	AL2 (CONDCOD1)	NORMAL CONDITION CODE 11.0	06670000	
M4CCODE2	DC	AL2 (CONDCOD2)	ERROR CONDITION CODE 11.0	06680000	
M4CCODE3	DC	AL2 (CONDCOD3)	NO SORT (SPECIFIED) 11.0	06690000	
M4CCODE4	DC	AL2 (CONDCOD4)	NO SORT (INVALID REQUESTS) 11.0	06700000	
M4MSDEC	DC	AL1 (DECMISOPT)	DECODE/COMPILE MSG OPTION X054	06710000	
M4MSPRO	DC	AL1 (PROMSOPT)	PROCESSING MSG OPTION X054	06720000	
M4MSRPT	DC	AL1 (RPTMSOPT)	REPORTING MSG OPTION X054	06730000	
	DC	X'00'	***** UNUSED ***** Z019	06740000	
M4WIDTH	DC	AL2 (LSTWIDTH)	M4LIST WIDTH (LRECL-1) Z019	06750000	
M4DEFWD	DC	AL2 (LSTDFWOP)	DEFAULT MK4 WIDTH-OF-PAGE Z019	06760000	
M4ADEFW	DC	AL2 (ALTDWOP)	DEFAULT ALT WIDTH-OF-PAGE Z019	06770000	
M4MAXGMN	DC	AL4 (MAXGETMN)	DEFAULT MAX GETMAIN Z021	06780000	
	DC	XL8'00'	***** UNUSED ***** Z019	06790000	
M4PAREND	EQU	*-1	### M4PARAMS END LOC ### QN10	06800000	
*				Z019	06810000
					06820000
END					

**M4LEPARM Source Code**

```

      TITLE 'VISION:UILDER LANGUAGE ENVIRONMENT PARAMETERS' 00010000
      ISEQ 73,80 00020000
* 00030000
***** 00040000
* 00050000
* PROPRIETARY AND CONFIDENTIAL INFORMATION OF 00060000
* COMPUTER ASSOCIATES INTERNATIONAL, INC. 00070000
* USE RESTRICTED BY WRITTEN LICENSE AGREEMENT 00080000
* 00090000
* DO NOT REMOVE THIS NOTICE 00100000
* 00110000
* 00120000
* COPYRIGHT (C) COMPUTER ASSOCIATES INTERNATIONAL, INC. 00130000
* AS AN UNPUBLISHED WORK. ALL RIGHTS RESERVED. 00140000
* 00150000
***** 00160000
M4LEPARM CSECT 00170000
***** 00180000
* 00190000
* THIS ROUTINE CONTAINS THE PARAMETERS USED TO INITIATE THE 00200000
* LANGUAGE ENVIRONMENT FOR VISION:UILDER. 00210000
* 00220000
* THESE PARAMETERS MAY BE MODIFIED AS APPROPRIATE FOR YOUR 00230000
* INSTALLATION REQUIREMENTS. THE PARAMETERS MUST CONFORM TO THE 00240000
* THE PARAMETERS DEFINED IN THE LANGUAGE ENVIRONMENT PROGRAMMING 00250000
* REFERENCE MANUAL FROM IBM. 00260000
* 00270000
* THIS ROUTINE MAY BE ASSEMBLED AND LINK EDITED AFTER BUILDER 00280000
* INSTALLATION IS COMPLETE. IF ALL OF THE DEFAULT PARAMETERS 00290000
* ARE SATISFACTORY, NO ACTION IS NEEDED. OTHERWISE, THE MODIFIED 00300000
* MODULE MUST BE ASSEMBLED AND LINK EDITED ACCORDING TO THE 00310000
* INSTRUCTIONS PROVIDED IN THE INSTALLATION MANUAL. 00320000
* 00330000
***** 00340000
      EJECT 00350000
* 00360000
* DO NOT CHANGE OR REMOVE THE FOLLOWING STATEMENTS. 00370000
* 00380000
      DC A(M4LEPLEN) 00390000
M4LEPLEN DC AL2(M4LEPEND-*-2) 00400000
      DC C'TRAP(OFF),' 00410000
* 00420000
* CHANGES MAY BE MADE TO THE FOLLOWING STATEMENTS AS APPROPRIATE. 00430000
* REFER TO THE IBM LANGUAGE ENVIRONMENT PROGRAMMING REFERENCE MANUAL 00440000
* FOR INFORMATION REGARDING APPLICABLE PARAMETERS AND THEIR MEANING. 00450000
* PARAMETERS MAY BE MODIFIED, REMOVED OR NEW PARAMETERS ADDED. 00460000
* INSTALLATION DEFAULTS WILL BE USED FOR ANY LANGUAGE ENVIRONMENT 00470000
* PARAMETER NOT SPECIFIED BELOW. THE FIRST OCCURRENCE OF THE '/' 00480000
* CHARACTER SIGNALS THE END OF THE LE PARAMETERS. TO TURN ON THE 00490000
* LE REPORTING FEATURES, CHANGE THE '/' TO A ',' AT THE END OF THE 00500000
* LINE BELOW DESIGNATED BY THE <----- INDICATOR. 00510000
* 00520000
      DC C'ALL31(OFF),' 00530000
      DC C'STACK(016K,016K,BELOW,KEEP),' 00540000
      DC C'LIBSTACK(04K,04K,FREE),' 00550000
      DC C'HEAP(008K,032K,ANY,KEEP,04K,04K),' 00560000
      DC C'ANYHEAP(016K,032K,ANY,KEEP),' 00570000
      DC C'BELOWHEAP(04K,04K,FREE),' 00580000
      DC C'THREADHEAP(04K,04K,ANY,KEEP),' 00590000
      DC C'MSGFILE(M4LEOUT,FBA,121,0) '/' <----- 00600000
      DC C'RPTOPTS(ON),RPTSTG(ON) '/' 00610000
* 00620000
* THE FOLLOWING STATEMENTS MUST NOT BE CHANGED. 00630000
* 00640000
M4LEPEND EQU * 00650000
* 00660000
      END 00670000

```

## M4SFPARM

M4SFPARM is a special parameter module, similar to M4PARAMS, that is used to define parameters for VISION:Builder options such as additional data validation symbols and changing automatic date validation formats. The basic module is supplied with VISION:Builder.

- Define parameters by adding statements to the source module, assembling the modified module, and link editing as with M4PARAMS.

The original source for M4SFPARM is delivered with four (4) additional user-defined validation symbols. These symbols and their character sets are described in the M4SFPARM source module. See [M4SFPARM on page B-19](#) for details.

- Define or change additional symbols by inserting statements into the M4SFPARM source module. Create these symbols using macros that you design to simplify the parameter specification. Define all the required macros at the beginning of the source module.
- Insert additional statements where indicated in the original source module. Insert them in any order within the limits that are defined.

### Defining Additional Data Validation Symbols

Each user-defined validation category consists of a validation symbol followed by the validation set definition. Categories are transcribed in the following format

Column	Entry	Description
10 – 14	M4SYM	This is the name of the macro that is used to define additional validation sets.
15	blank	

Column	Entry	Description
16		A 1-character entry to be used as a validation symbol.  Each user-defined validation symbol must be unique; it cannot be one of the standard validation symbols or a previously-defined user validation symbol. It can be any character other than the minus sign (-), apostrophe ('), underscore (_), bar ( ), slash (/), ampersand (&), comma (,), blank ( ), the numbers 0 through 8, and the VISION:Builder system delimiter set in M4PARAMS.
17	comma	
18– 71		Set definition, bounded by apostrophes.  Any and all characters can be included in the set definition. However, if an apostrophe is to be an element of the set, it must appear in two consecutive columns (see <a href="#">Figure 2-1</a> ). The same requirement also applies to the ampersand.

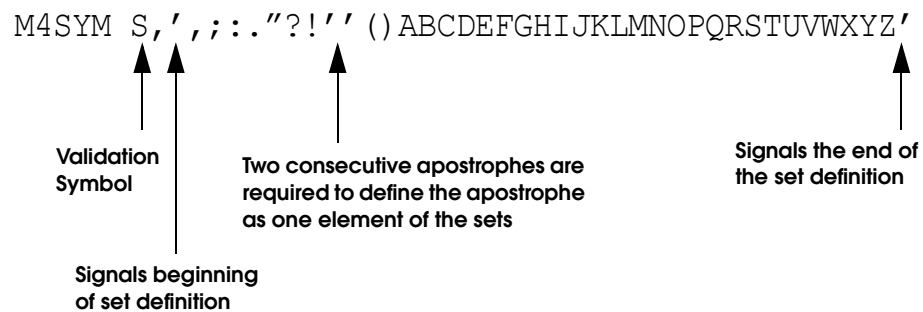


Figure 2-1 Sample Validation Category Definition

## Change Automatic Date Validation Format

This capability is used to change the format of the date. The standard format is MMDDYY, six digits specifying month, day, and year.

A non-standard date format is specified with a statement in the following format.

**Note:** If your standard date format is changed in M4PARAMS, it must also be changed here if the DV operator (date validation) is to validate a date according to your installation standards.

Column	Entry	Description
10 – 14	M4DAT	
15	blank	
16 – 21		<p>These columns are used to specify the relative position of a date's components.</p> <p>The <i>day</i> position is represented by the two characters DD, <i>month</i> by MM, and the <i>year</i> by YY. The D, M, and Y characters must always appear in pairs, with each pair being specified exactly once. Following are the possible legal permutations of the date:</p> <p>MMDDYY  MMYYDD  DDMMYY  DDYMM  YYDDMM  YYMMDD</p>

## M4SFARM

```

MF          TITLE 'M4SFARM - COMPUTER ASSOCIATES INTERNATIONAL, INC.'      00010000
           ISEQ 73,80                                                         00020000
*****
*          * 00030000
*          * 00040000
*          * 00050000
*          * 00060000
*          * 00070000
*          * 00080000
*          * 00090000
*          * 00100000
*          * 00110000
*          * 00120000
*          * 00130000
*          * 00140000
*****
*          * 00150000
*          * 00160000
*          * 00170000
*          * 00180000
*          * 00190000
*          * 00200000
*          * 00210000
*          * 00220000
*          * 00230000
*          * 00240000
*          * 00250000
*          * 00260000
*          * 00270000
*          * 00280000
*          * 00290000
*          * 00300000
*          * 00310000
*          * 00320000
*          * 00330000
*          * 00340000
*          * 00350000
*          * 00360000
*          * 00370000
*****
MACRO
M4SYM &SETNAME,&ELEMENT
LCLA &INDEX
LCLA &NE
LCLC &SN
LCLC &AMPER
&AMPER SETC '&'(1,1)
&INDEX SETA K'&ELEMENT
&NE SETA K'&ELEMENT-2
&SN SETC '&SETNAME'
DC CL6'SYMBOL'
AIF (K'&SETNAME EQ 1).SF10
MNOTE 'ILLEGAL VALIDATION SYMBOL LENGTH'
&SN SETC ' '
.SF10 ANOP
DC CL1'&SN.'
AIF (&NE GT 0).SF20
MNOTE 'NO SET ELEMENTS DEFINED'
SPACE 3
MEXIT
.SF20 ANOP
AIF (&INDEX LT 3).SF40

```

**M4SFPARM (cont.)**

```

&INDEX   SETA   &INDEX-1                                00380000
          AIF    ('&ELEMENT'(&INDEX,1) EQ '&AMPER').SF30    00390000
          AIF    ('&ELEMENT'(&INDEX,1) NE ' ').SF20         00400000
.SF30     ANOP                                     00410000
&NE       SETA   &NE-1                                00420000
&INDEX    SETA   &INDEX-1                              00430000
          AGO     .SF20                                00440000
.SF40     ANOP                                     00450000
          DC      FL1 '&NE.'                          00460000
          DC      CL&NE.&ELEMENT                      00470000
          SPACE   3                                    00480000
          MEND                                         00490000
          MACRO                                       00500000
          M4DAT   &DATEFMT                            00510000
          LCLC    &DFMT                               00520000
&DFMT     SETC   '&DATEFMT'                          00530000
          AIF    (K'&DATEFMT EQ 6).SF100              00540000
          MNOTE   'ILLEGAL DATE FORMAT'               00550000
          SPACE   3                                    00560000
          MEXIT                                       00570000
.SF100    ANOP                                     00580000
          DC      CL4 'DATE'                          00590000
          DC      CL6 '&DFMT.'                        00600000
          SPACE   3                                    00610000
          MEND                                         00620000
          MACRO                                       00630000
          M4END                                         00640000
          DC      CL12 'END M4SFPARM'                  00650000
          MEND                                         00660000
M4SFPARM  START                                       00670000
*                                                00680000
*          NOTE: OPERAND OF START INSTRUCTION MUST REMAIN BLANK 00690000
*                                                00700000
* - - - - -                                           00710000
* THE FOLLOWING USER-DEFINED SYMBOLS AND CHARACTER SETS ARE DEFINED 00720000
* HERE FOR USE IN PATTERN VALIDATION OPERATIONS. 00730000
*                                                00740000
* 'a' - UPPER & LOWER CASE ALPHA (A-Z,a-z) OR BLANK 00750000
* 'x' - UPPER & LOWER CASE ALPHA (A-Z,a-z), NUMERIC (0-9) OR BLANK 00760000
* 'y' - UPPER & LOWER CASE ALPHA (A-Z,a-z) OR NUMERIC (0-9) 00770000
* 'z' - UPPER & LOWER CASE ALPHA (A-Z,a-z) 00780000
*                                                00790000
          M4SYM a, 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz X00800000
          '                                           00810000
          M4SYM x, 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0X00820000
          123456789 ' 00830000
          M4SYM y, 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0X00840000
          123456789' 00850000
          M4SYM z, 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz' 00860000
* - - - - -                                           00870000
*                                                00880000
***** 00890000
* INSERT ADDITIONAL STATEMENTS AFTER THIS STATEMENT 00900000
* DO NOT INSERT STATEMENTS AFTER THIS STATEMENT 00910000
***** 00920000
          M4END                                         00930000
          END                                           00940000

```

**Example of Additional Source Statements**

[Figure 2-2](#) shows a list of additional source statements that define three new validation symbols and change the format of the date.

```

M4SYM T, '¢$.+--0123456789'
M4SYM U, '+-*/='
M4SYM V, 'AEFIJOP'
M4DAT YMMDD

```

Figure 2-2 Sample Additional Statements

## MARKLIBP

COMLIB provides standard default conditions for the library parameters usually determined by the operating environment in each installation. These parameters affect directory blocking, optional library tracking information, object compression, and reserve. Because these parameters are functions of the installation, COMLIB provides you with the capability of changing the default parameters.

A special program module called MARKLIBP is used for this purpose and is supplied with COMLIB. It is made available to you and can be changed to suit your needs. Each condition supplied as the default standard is shown. The module itself is supplied as an assembly language source CSECT and is well documented in its source form. A listing of the CSECT for OS/390 (MVS) follows; the default is clearly indicated for each parameter.

Parameter	Parameter Name	COMLIB Default	
Library directory blocking factor	DIRBLK	0 - allows COMLIB to calculate the optimum directory blocking factor based on the device and size of the library.	
Not applicable to VSAM		A larger value improves decode and library maintenance performance. The table shown indicates maximum values depending on device.	
		Device	Maximum Blocking Factor
		3380	733
		3390	800
COMLIB reserve flag for shared DASD	RESERVE	0	
Item tracking flag	ITEMTRAK	0	
Minimum Compress Size	MINCMPSZ	507	
Compression Flag	COMPRESS	0	

## MARKLIBP SOURCE

```

PBLP4      TITLE 'MARKLIBP  - COMPUTER ASSOCIATES INTERNATIONAL, INC.'      00010000
          ISEQ  73,80                                                         00020000
*****
*                                                         * 00030000
*                                                         * 00040000
*          PROPRIETARY AND CONFIDENTIAL INFORMATION OF      * 00050000
*          COMPUTER ASSOCIATES INTERNATIONAL, INC.          * 00060000
*          USE RESTRICTED BY WRITTEN LICENSE AGREEMENT      * 00070000
*                                                         * 00080000
*          DO NOT REMOVE THIS NOTICE                        * 00090000
*                                                         * 00100000
*          COPYRIGHT (C) COMPUTER ASSOCIATES INTERNATIONAL, INC. * 00110000
*          AS AN UNPUBLISHED WORK.  ALL RIGHTS RESERVED.    * 00120000
*                                                         * 00130000
*                                                         * 00140000
*          NOTE: U.S. GOVERNMENT DFARS CONTRACTS, RESTRICTED RIGHTS LEGEND: * 00150000
*          USE, DUPLICATION, OR DISCLOSURE IS SUBJECT TO RESTRICTIONS * 00160000
*          STATED IN PARAGRAPH (C) (1) (II) OF THE RIGHTS IN TECHNICAL DATA * 00170000
*          AND COMPUTER SOFTWARE CLAUSE AT DFARS 252.227-7013. * 00180000
*                                                         * 00190000
*****
          MACRO                                                         00200000
          DIRBLK &FACTOR                                                         00210000
DIRBLKSZ EQU  &FACTOR*32                                                         00220000
          MEND                                                         00230000
*                                                         00240000
MARKLIBP CSECT                                                         00250000
*                                                         00260000
*                                                         00270000
*DIRECTORY SPECIFIES THE NUMBER OF ENTRIES IN A LIBRARY DIRECTORY 00280000
*BLOCKING BLOCK.  LARGER VALUES DECREASE ACCESS TIME FOR LARGE 00290000
*FACTOR LIBRARIES.  THE FACTOR CANNOT EXCEED THE VALUES GIVEN 00300000
* BELOW FOR THE DEVICE USED.  A VALUE OF 0 ALLOWS COMLIB 00310000
* TO CALCULATE THE OPTIMUM DIRECTORY BLOCKING FACTOR 00320000
* BASED ON THE DEVICE AND SIZE OF THE LIBRARY. 00330000
* NOT APPLICABLE TO VSAM LIBRARIES. 00340000
*                                                         00350000
*          3330      372                                                         00360000
*          3340      243                                                         00370000
*          3350      561                                                         00380000
*          3375      549                                                         00390000
*          3380      733                                                         00400000
*          3390      800 (ALSO APPLIES TO HITACHI DEVICE H6587) 00410000
*          9345      692                                                         00420000
*                                                         00430000
          DIRBLK 0                                                         00440000
*                                                         00450000
*DEVICE THE DEVICE RESERVE FLAG PROVIDES THE ABILITY TO ISSUE 00460000
*RESERVE A DEVICE RESERVE RATHER THAN AN OPERATING SYSTEM ENQUE. 00470000
*FLAG ENTER A '1' TO ACTIVATE THE RESERVE FEATURE FOR A 00480000
* MAXIMUM OF ONE LIBRARY IN A GIVEN RUN.  ENTER A 2 TO 00490000
* ACTIVATE THE MULTILIB RESERVE FEATURE WHICH SUPPORTS 00500000
* DEVICE RESERVATION FOR MORE THAN ONE LIBRARY IN A 00510000
* GIVEN RUN. (NOT APPLICABLE TO DOS/VS(E)) 00520000
*                                                         00530000
RESERVE EQU 0                                                         00540000
*                                                         00550000
*ITEM THE ITEM TRACKING FLAG ENABLES OPTIONAL FEATURES OF THE 00560000
*TRACKING ITEM TRACKING FACILITY.  ENTER A '1' TO REQUIRE THE USE 00570000
*FLAG OF IT STATEMENTS WITH UPDATER IDENTIFICATION FOR ALL 00580000
* LIBRARY UPDATES.  ENTER A '2' TO ENABLE THE DATE-OF-ITEM 00590000
* USE FACILITY.  ENTER A '3' TO ENABLE BOTH. 00600000
*                                                         00610000
ITEMTRAK EQU 0                                                         00620000
*                                                         C011 00630000
*                                                         C011 00640000
*****C011 00650000
*****C011 00660000
* THE FOLLOWING ENTRIES HAVE TO DO WITH THE OPTIONAL COMPRESSION *C011 00670000
* OF OBJECTS STORED IN THE LIBRARY.  COMPRESSED OBJECTS MUST BE *C011 00680000
* DE-COMPRESSED WHEN THEY ARE RETRIEVED.  THIS WILL ADD SOME *C011 00690000

```

**MARKLIBP SOURCE (cont.)**

```

* MINIMAL OVERHEAD IN EXCHANGE FOR REDUCED I/O COUNTS AND DISK *C011 00700000
* SPACE USAGE. COMPRESSION WILL BE MOST ADVANTAGEOUS IF YOUR *C011 00710000
* LIBRARY OBJECTS TEND TO BE LARGE (MORE THAN 100 FIELDS IN A *C011 00720000
* FILE DEFINITION OR 200 ENTRIES IN A TABLE). IF COMPRESSED *C011 00730000
* OBJECTS HAVE BEEN STORED IN THE LIBRARY, THEY WILL STILL BE *C011 00740000
* RETRIEVED AND DE-COMPRESSED EVEN THOUGH LIBRARY COMPRESSION *C011 00750000
* WAS SUBSEQUENTLY DISABLED. A LIBRARY WITH COMPRESSED OBJECTS *C011 00760000
* MAY BE CONVERTED TO A LIBRARY WITHOUT ANY COMPRESSED OBJECTS *C011 00770000
* BY DUMPING THE LIBRARY AND THEN RESTORING IT WITH COMPRESSION *C011 00780000
* DISABLED. *C011 00790000
* *C011 00800000
* THE COMPRESSION SOFTWARE IS PROVIDED COURTESY OF *C011 00810000
* JEAN-LOUP GAILLY AND MARK ADLER. *C011 00820000
*****C011 00830000
* C011 00840000
*MINIMUM THE MINIMUM COMPRESS SIZE SPECIFIES THE MINIMUM C011 00850000
*COMPRESS SIZE THAT A LIBRARY OBJECT MUST BE BEFORE IT WILL C011 00860000
*SIZE BE COMPRESSED WHENEVER COMPRESSION IS ENABLED. C011 00870000
* OBJECTS LARGER THAN THE SPECIFIED SIZE WILL BE C011 00880000
* COMPRESSED WHENEVER COMPRESSION IS ENABLED. C011 00890000
* OBJECTS WHOSE SIZE IS LESS THAN OR EQUAL TO THE C011 00900000
* SPECIFIED SIZE WILL NOT BE COMPRESSED REGARDLESS C011 00910000
* OF WHETHER COMPRESSION IS ENABLED OR NOT. C011 00920000
* THE SPECIFIED SIZE SHOULD NEVER BE SMALLER THAN C011 00930000
* 507 AND MAY BE SPECIFIED AS LARGE AS 65535. C011 00940000
* C011 00950000
MINCMPSZ EQU 507 C011 00960000
* C011 00970000
*COMPRESSION THE COMPRESSION FLAG ENABLES THE OPTIONAL LIBRARY C011 00980000
*FLAG OBJECT COMPRESSION. ENTER A '1' TO ENABLE THE C011 00990000
* COMPRESSION OF LIBRARY OBJECTS WHOSE SIZE EXCEEDS C011 01000000
* THE 'MINIMUM COMPRESS SIZE' SPECIFIED ABOVE. C011 01010000
* ENTER A '0' TO DISABLE LIBRARY OBJECT COMPRESSION. C011 01020000
* C011 01030000
COMPRESS EQU 0 C011 01040000
* C011 01050000
* 01060000
***** 01070000
* USERS MUST NOT MAKE CHANGES FOLLOWING THIS PAGE 01080000
***** 01090000
EJECT 01100000
ORG 01110000
LPRELNO DC CL4'C4.5' 01120000
LPDIRBLK DC Y(DIRBLKSZ) 01130000
LPRESERV DC AL1(RESERVE) 01140000
LPITKFLG DC AL1(ITEMTRAK) 01150000
LPCMPSIZ DC AL2(MINCMPSZ) C011 01160000
LPCMPFLG DC AL1(COMPRESS) C011 01170000
* C011 01180000
END 01190000

```

**MARKSQL****MARKSQL**

```

SQL TITLE 'SQL STATEMENT GENERATOR FOR BUILDER-DB2 INTERFACE' 00010000
***** 00020000
* 00030000
* PROPRIETARY AND CONFIDENTIAL INFORMATION OF * 00040000
* COMPUTER ASSOCIATES INTERNATIONAL, INC. * 00050000
* USE RESTRICTED BY WRITTEN LICENSE AGREEMENT * 00060000
* 00070000
* DO NOT REMOVE THIS NOTICE * 00080000
* 00090000
* 00100000

```

## MARKSQL (cont.)

```

*          COPYRIGHT (C) COMPUTER ASSOCIATES INTERNATIONAL, INC.          * 00110000
*          AS AN UNPUBLISHED WORK. ALL RIGHTS RESERVED.                   * 00120000
*                                                                           * 00130000
*****                                                                           * 00140000
*                                                                           * 00150000
* THIS MODULE GENERATES THE DYNAMIC SQL STATEMENTS USED BY BUILDER        * 00160000
* TO INTERFACE WITH DB2. THE GLOBAL SET SYMBOL 'MAX' DEFINED BELOW        * 00170000
* MUST BE SET TO THE MAXIMUM NUMBER OF DB2 TABLES WHICH WILL BE        * 00180000
* ACCESSED WITHIN ANY BUILDER APPLICATION. AFTER SETTING THE 'MAX'      * 00190000
* VALUE APPROPRIATELY, THIS PROGRAM SHOULD BE ASSEMBLED TO GENERATE     * 00200000
* THE BUILDER-DB2 INTERFACE PROGRAM STATEMENTS. THIS GENERATED         * 00210000
* PROGRAM (THE PUNCH OUTPUT FROM THE ABOVE ASSEMBLY STEP) MUST THEN      * 00220000
* BE PROVIDED AS INPUT TO THE DB2 PRE-PROCESSOR PROGRAM FOR              * 00230000
* PREPARATION OF THE BUILDER APPLICATION PLAN TO BE USED TO ACCESS       * 00240000
* YOUR DB2 DATA BASES. SEE YOUR BUILDER INSTALLATION GUIDE FOR         * 00250000
* FURTHER EXPLANATION REGARDING THE FUNCTION OF THIS MODULE.             * 00260000
*                                                                           * 00270000
* FOLLOWING IS SOME SAMPLE JCL FOR THE ASSEMBLY AND PROGRAM              * 00280000
* PREPARATION STEPS REQUIRED FOR THIS PROGRAM FOR USE IN THE              * 00290000
* TSO/BATCH ENVIRONMENT.                                                 * 00300000
*                                                                           * 00310000
*          //JOBNAME JOB ...                                             * 00320000
*          //*                                                             * 00330000
*          //GEN EXEC PGM=ASMA90,PARM='DECK,NOOBJECT',REGION=2M          * 00340000
*          //SYSPRINT DD SYSOUT=*                                         * 00350000
*          //SYSPUNCH DD DSN=&&GENOUT,DISP=(MOD,PASS),UNIT=SYSDA,         * 00360000
*          //          SPACE=(800,(200,200)),                             * 00370000
*          //          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)              * 00380000
*          //SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))                       * 00390000
*          //SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR                           * 00400000
*          //SYSIN DD DSN=THIS.PROGRAM.SOURCE,DISP=SHR                   * 00410000
*          //*                                                             * 00420000
*          //SQLPREP EXEC DSNHASM,MEM=MARKIV,USER=YOURID,                 * 00430000
*          //          PARM.PC='HOST(ASM),STDSQL(86)'                     * 00440000
*          //PC.SYSIN DD DSN=&&GENOUT,DISP=(OLD,DELETE)                    * 00450000
*          //LKED.SYSLMOD DD DSN=YOUR.MARKIV.LOADLIB,DISP=OLD             * 00460000
*          //LKED.SYSIN DD *                                               * 00470000
*          INCLUDE SYSLIB(DSNELI)                                         * 00480000
*          MODE AMODE(31),RMODE(ANY)                                       * 00490000
*          NAME MARKSQLT(R)                                                * 00500000
*          //                                                             * 00510000
*                                                                           * 00520000
* IF YOU WISH TO USE THE CALL ATTACHMENT FACILITY IN PLACE OF OR         * 00530000
* IN ADDITION TO THE TSO ATTACHMENT FACILITY (WHICH REQUIRES THE        * 00540000
* TSO TERMINAL MONITOR PROGRAM AND THE DSN COMMAND PROCESSOR), THE      * 00550000
* THE LINKAGE EDITOR SYSIN STATEMENTS IN THE ABOVE EXAMPLE SHOULD      * 00560000
* BE CHANGED AS FOLLOWS:                                                 * 00570000
*                                                                           * 00580000
*          INCLUDE SYSLIB(DSNALI)                                         * 00590000
*          MODE AMODE(31),RMODE(ANY)                                       * 00600000
*          NAME MARKSQLC(R)                                                * 00610000
*                                                                           * 00620000
*          AND THE "MEM=" NAME SHOULD BE CHANGED TO MARKDB2.             * 00630000
*                                                                           * 00640000
* FOR THE IMS ENVIRONMENT, THE IMS ATTACHMENT FACILITY IS REQUIRED        * 00650000
* AND THE LINKAGE EDITOR SYSIN STATEMENTS IN THE ABOVE EXAMPLE         * 00660000
* MUST BE CHANGED AS FOLLOWS:                                           * 00670000
*                                                                           * 00680000
*          INCLUDE SYSLIB(DFSLLI000)                                       * 00690000
*          MODE AMODE(31),RMODE(ANY)                                       * 00700000
*          NAME MARKSQLI(R)                                                * 00710000
*                                                                           * 00720000
*          AND THE "MEM=" NAME SHOULD BE CHANGED TO MARKDLI.             * 00730000
*                                                                           * 00740000
*                                                                           * 00750000
* NOTE THAT ALL THREE ATTACHMENT FACILITY INTERFACE PROGRAMS MAY        * 00760000
* BE PREPARED AND PLACED INTO THE BUILDER LOAD LIBRARY. BUILDER        * 00770000
* WILL SELECT THE APPROPRIATE PROGRAM BASED UPON THE EXECUTION          * 00780000
* ENVIRONMENT.                                                            * 00790000
*                                                                           * 00800000
* ALSO NOTE THAT THE BUILDER-DB2 INTERFACE PROGRAM WILL BE ENTERED     * 00810000

```

## MARKSQL (cont.)

```

* IN THE 31-BIT ADDRESSING MODE.  THEREFORE, THIS PROGRAM MAY BE      * 00820000
* LINK EDITED WITH THE 'MODE AMODE(31),RMODE(ANY)' LINKAGE EDITOR      * 00830000
* CONTROL STATEMENT TO ALLOW THE PROGRAM TO BE LOADED INTO            * 00840000
* VIRTUAL STORAGE ABOVE THE 16MB LINE FOR MVS/XA OR MVS/ESA            * 00850000
* OPERATING SYSTEMS.                                                  * 00860000
*                                                                      * 00870000
*****                                                                * 00880000
*                                                                      * 00890000
          GBLA  &N,&MAX                                              * 00900000
*                                                                      * 00910000
*****                                                                * 00920000
*                                                                      * 00930000
&MAX      SETA  100          MAXIMUM NUMBER OF SQL STATEMENTS PER APPL. * 00940000
*                                                                      * 00950000
*****                                                                * 00960000
*                                                                      * 00970000
*  !!NOTE!!  DO NOT CHANGE ANY STATEMENTS BEYOND THIS LINE.  !!NOTE!! * 00980000
*                                                                      * 00990000
*****                                                                * 01000000
          EJECT                                                    * 01010000
          PUNCH '          SPACE 3'                                * 01020000
          PUNCH '          MACRO'                                  * 01030000
          PUNCH '&&LABEL      SQLENTER &&AREA'                      * 01040000
          PUNCH '          AIF  ('&&LABEL' EQ ''').NOLABEL'        * 01050000
          PUNCH '&&LABEL      DS    0H'                              * 01060000
          PUNCH '.NOLABEL  ANOP'                                    * 01070000
          PUNCH '          USING *,9'                              * 01080000
          PUNCH '          AIF  ('&&AREA' EQ ''').NOAREA'          * 01090000
          PUNCH '          USING &&AREA,5'                          * 01100000
          PUNCH '.NOAREA  ANOP'                                    * 01110000
          PUNCH '          MEND'                                    * 01120000
          PUNCH '          SPACE 3'                                * 01130000
          SPACE 3                                                  * 01140000
          PUNCH '          MACRO'                                  * 01150000
          PUNCH '&&LABEL      SQLRET'                                * 01160000
          PUNCH '          AIF  ('&&LABEL' EQ ''').NOLABEL'        * 01170000
          PUNCH '&&LABEL      DS    0H'                              * 01180000
          PUNCH '.NOLABEL  ANOP'                                    * 01190000
          PUNCH '          BR    8'                                * 01200000
          PUNCH '          MEND'                                    * 01210000
          PUNCH 'SQL      TITLE  'SQL STATEMENTS FOR BUILDER-DB2'' * 01220000
          EJECT                                                    * 01230000
          PUNCH 'MARKSQL  CSECT'                                    * 01240000
          PUNCH 'MARKSQL  AMODE 31'                                * 01250000
          PUNCH 'MARKSQL  RMODE ANY'                              * 01260000
          PUNCH '          USING SQLCA,2'                          * 01270000
          PUNCH '          USING SQLDSECT,3'                      * 01280000
          PUNCH '          USING SQLCODEX,6'                      * 01290000
          PUNCH '          USING STMT,4'                          * 01300000
          PUNCH '*'                                                * 01310000
          PUNCH '          DC    CL8'MARKSQL1'                    * 01320000
          PUNCH '*'                                                * 01330000
          PUNCH '          DC    A(SQLDLEN)                        * 01340000
          PUNCH '*'                                                * 01350000
          PUNCH '*  VECTORS TO LIST OF INDIVIDUAL STATEMENT VECTORS' * 01360000
          PUNCH '*'                                                * 01370000
          PUNCH '          DC    A(OPENLIST)'                      * 01380000
          PUNCH '          DC    A(OPUDLIST)'                     * 01390000
          PUNCH '          DC    A(CLOSLIST)'                      * 01400000
          PUNCH '          DC    A(PREPLIST)'                      * 01410000
          PUNCH '          DC    A(DESCLIST)'                     * 01420000
          PUNCH '          DC    A(FTCHLIST)'                     * 01430000
          PUNCH '          DC    A(EXECLIST)'                      * 01440000
          PUNCH '          DC    A(XECILIST)'                      * 01450000
          PUNCH '          DC    A(COMWLIST)'                      * 01460000
          PUNCH '          DC    A(INTOLIST)'                      * 01470000
          SPACE 3                                                  * 01480000
          PUNCH '*'                                                * 01490000
          PUNCH '*  SQL OPEN CURSOR STATEMENT VECTOR LIST'        * 01500000
          PUNCH '*'                                                * 01510000
          PUNCH 'OPENLIST EQU  *'                                  * 01520000

```

## MARKSQL (cont.)

	PUNCH	'	DC	A((OPENEND-*)/4-1)'	01530000
&N	SETA	1			01540000
.OPENV	ANOP				01550000
	PUNCH	'	DC	A(OPEN&N)'	01560000
&N	SETA	&N+1			01570000
	AIF	(&N LE &MAX).	OPENV		01580000
	PUNCH	'OPENEND EQU	*		01590000
	SPACE	3			01600000
	PUNCH	'**'			01610000
	PUNCH	'* SQL OPEN CURSOR USING DESCRIPTOR STMT VECTOR LIST'			01620000
	PUNCH	'**'			01630000
	PUNCH	'OPUDLIST EQU	*		01640000
	PUNCH	'	DC	A((OPUDEND-*)/4-1)'	01650000
&N	SETA	1			01660000
.OPUDV	ANOP				01670000
	PUNCH	'	DC	A(OPUD&N)'	01680000
&N	SETA	&N+1			01690000
	AIF	(&N LE &MAX).	OPUDV		01700000
	PUNCH	'OPUDEND EQU	*		01710000
	SPACE	3			01720000
	PUNCH	'**'			01730000
	PUNCH	'* SQL CLOSE CURSOR STATEMENT VECTOR LIST'			01740000
	PUNCH	'**'			01750000
	PUNCH	'CLOSLIST EQU	*		01760000
	PUNCH	'	DC	A((CLOSEND-*)/4-1)'	01770000
&N	SETA	1			01780000
.CLOSV	ANOP				01790000
	PUNCH	'	DC	A(CLOSE&N)'	01800000
&N	SETA	&N+1			01810000
	AIF	(&N LE &MAX).	CLOSV		01820000
	PUNCH	'CLOSEND EQU	*		01830000
	SPACE	3			01840000
	PUNCH	'**'			01850000
	PUNCH	'* SQL PREPARE STATEMENT VECTOR LIST'			01860000
	PUNCH	'**'			01870000
	PUNCH	'PREPLIST EQU	*		01880000
	PUNCH	'	DC	A((PREPEND-*)/4-1)'	01890000
&N	SETA	1			01900000
.PREPV	ANOP				01910000
	PUNCH	'	DC	A(PREP&N)'	01920000
&N	SETA	&N+1			01930000
	AIF	(&N LE &MAX).	PREPV		01940000
	PUNCH	'PREPEND EQU	*		01950000
	SPACE	3			01960000
	PUNCH	'**'			01970000
	PUNCH	'* SQL DESCRIBE STATEMENT VECTOR LIST'			01980000
	PUNCH	'**'			01990000
	PUNCH	'DESCLIST EQU	*		02000000
	PUNCH	'	DC	A((DESCEND-*)/4-1)'	02010000
&N	SETA	1			02020000
.DESCV	ANOP				02030000
	PUNCH	'	DC	A(DESC&N)'	02040000
&N	SETA	&N+1			02050000
	AIF	(&N LE &MAX).	DESCV		02060000
	PUNCH	'DESCEND EQU	*		02070000
	SPACE	3			02080000
	PUNCH	'**'			02090000
	PUNCH	'* SQL FETCH STATEMENT VECTOR LIST'			02100000
	PUNCH	'**'			02110000
	PUNCH	'FTCHLIST EQU	*		02120000
	PUNCH	'	DC	A((FTCHEND-*)/4-1)'	02130000
&N	SETA	1			02140000
.FTCHV	ANOP				02150000
	PUNCH	'	DC	A(FTCH&N)'	02160000
&N	SETA	&N+1			02170000
	AIF	(&N LE &MAX).	FTCHV		02180000
	PUNCH	'FTCHEND EQU	*		02190000
	SPACE	3			02200000
	PUNCH	'**'			02210000
	PUNCH	'* SQL EXECUTE STATEMENT VECTOR LIST'			02220000
	PUNCH	'**'			02230000

## MARKSQL (cont.)

	PUNCH 'EXECLIST EQU *'	02240000
	PUNCH ' DC A((EXECEND-*)/4-1)'	02250000
&N	SETA 1	02260000
.EXECV	ANOP	02270000
	PUNCH ' DC A(EXEC&N)'	02280000
&N	SETA &N+1	02290000
	AIF (&N LE &MAX).EXECV	02300000
	PUNCH 'EXECEND EQU *'	02310000
	SPACE 3	02320000
	PUNCH '**'	02330000
	PUNCH '* SQL EXECUTE IMMEDIATE STATEMENT VECTOR LIST'	02340000
	PUNCH '**'	02350000
	PUNCH 'XECILIST EQU *'	02360000
	PUNCH ' DC A(1)'	02370000
	PUNCH ' DC A(EXECIMMD)'	02380000
	SPACE 3	02390000
	PUNCH '**'	02400000
	PUNCH '* SQL COMMIT WORK STATEMENT VECTOR LIST'	02410000
	PUNCH '**'	02420000
	PUNCH 'COMWLIST EQU *'	02430000
	PUNCH ' DC A(1)'	02440000
	PUNCH ' DC A(COMMIT)'	02450000
	SPACE 3	A019 02460000
	PUNCH '**'	A019 02470000
	PUNCH '* SQL PREPARE INTO STATEMENT VECTOR LIST'	A019 02480000
	PUNCH '**'	A019 02490000
	PUNCH 'INTOLIST EQU *'	A019 02500000
	PUNCH ' DC A(1)'	A019 02510000
	PUNCH ' DC A(PREPINTO)'	A019 02520000
	PUNCH ' EJECT'	02530000
	EJECT	02540000
*		02550000
	PUNCH '**'	02560000
	PUNCH '* DECLARE CURSOR STATEMENT'	02570000
	PUNCH '**'	02580000
&N	SETA 1	02590000
.DECLS	ANOP	02600000
	PUNCH ' EXEC SQL DECLARE CUR&N CURSOR WITH HOLD FOR STMT&N'	02610000
	PUNCH '**'	02620000
&N	SETA &N+1	02630000
	AIF (&N LE &MAX).DECLS	02640000
	PUNCH ' EJECT'	02650000
	EJECT	02660000
*		02670000
	PUNCH '**'	02680000
	PUNCH '* OPEN CURSOR STATEMENT'	02690000
	PUNCH '**'	02700000
&N	SETA 1	02710000
.OPENS	ANOP	02720000
	PUNCH 'OPEN&N SQLENTER'	02730000
	PUNCH ' EXEC SQL OPEN CUR&N'	02740000
	PUNCH ' SQLRET'	02750000
	PUNCH ' LTORG'	02760000
	PUNCH '**'	02770000
&N	SETA &N+1	02780000
	AIF (&N LE &MAX).OPENS	02790000
	PUNCH ' EJECT'	02800000
	EJECT	02810000
*		02820000
	PUNCH '**'	02830000
	PUNCH '* OPEN CURSOR USING DESCRIPTOR STATEMENT'	02840000
	PUNCH '**'	02850000
&N	SETA 1	02860000
.OPUDS	ANOP	02870000
	PUNCH 'OPUD&N SQLENTER SQLDA&N'	02880000
	PUNCH ' EXEC SQL OPEN CUR&N USING DESCRIPTOR :SQLDA&N'	02890000
	PUNCH ' SQLRET'	02900000
	PUNCH ' LTORG'	02910000
	PUNCH '**'	02920000
&N	SETA &N+1	02930000
	AIF (&N LE &MAX).OPUDS	02940000

**MARKSQL (cont.)**

	PUNCH ' EJECT'	02950000
	EJECT	02960000
*		02970000
	PUNCH '*'	02980000
	PUNCH '* CLOSE CURSOR STATEMENT'	02990000
	PUNCH '*'	03000000
&N	SETA 1	03010000
.CLOSS	ANOP	03020000
	PUNCH 'CLOSE&N SQLENTER'	03030000
	PUNCH ' EXEC SQL CLOSE CUR&N'	03040000
	PUNCH ' SQLRET'	03050000
	PUNCH ' LTORG'	03060000
	PUNCH '*'	03070000
&N	SETA &N+1	03080000
	AIF (&N LE &MAX).CLOSS	03090000
	PUNCH ' EJECT'	03100000
	EJECT	03110000
*		03120000
	PUNCH '*'	03130000
	PUNCH '* PREPARE STATEMENT'	03140000
	PUNCH '*'	03150000
&N	SETA 1	03160000
.PREPS	ANOP	03170000
	PUNCH 'PREP&N SQLENTER'	03180000
	PUNCH ' EXEC SQL PREPARE STMT&N FROM :STMT'	03190000
	PUNCH ' SQLRET'	03200000
	PUNCH ' LTORG'	03210000
	PUNCH '*'	03220000
&N	SETA &N+1	03230000
	AIF (&N LE &MAX).PREPS	03240000
	PUNCH ' EJECT'	03250000
	EJECT	03260000
*		03270000
	PUNCH '*'	03280000
	PUNCH '* DESCRIBE STATEMENT'	03290000
	PUNCH '*'	03300000
&N	SETA 1	03310000
.DESCS	ANOP	03320000
	PUNCH 'DESC&N SQLENTER SQLDA&N'	03330000
	PUNCH ' EXEC SQL DESCRIBE STMT&N INTO :SQLDA&N'	03340000
	PUNCH ' SQLRET'	03350000
	PUNCH ' LTORG'	03360000
	PUNCH '*'	03370000
&N	SETA &N+1	03380000
	AIF (&N LE &MAX).DESCS	03390000
	PUNCH ' EJECT'	03400000
	EJECT	03410000
*		03420000
	PUNCH '*'	03430000
	PUNCH '* FETCH STATEMENT'	03440000
	PUNCH '*'	03450000
&N	SETA 1	03460000
.FTCHS	ANOP	03470000
	PUNCH 'FTCH&N SQLENTER SQLDA&N'	03480000
	PUNCH ' EXEC SQL FETCH CUR&N USING DESCRIPTOR :SQLDA&N'	03490000
	PUNCH ' SQLRET'	03500000
	PUNCH ' LTORG'	03510000
	PUNCH '*'	03520000
&N	SETA &N+1	03530000
	AIF (&N LE &MAX).FTCHS	03540000
	PUNCH ' EJECT'	03550000
	EJECT	03560000
*		03570000
	PUNCH '*'	03580000
	PUNCH '* EXECUTE STATEMENT'	03590000
	PUNCH '*'	03600000
&N	SETA 1	03610000
.EXECS	ANOP	03620000
	PUNCH 'EXEC&N SQLENTER SQLDA&N'	03630000
	PUNCH ' EXEC SQL EXECUTE STMT&N USING DESCRIPTOR :SQLDA&N'	03640000
	PUNCH ' SQLRET'	03650000

## MARKSQL (cont.)

	PUNCH '          LTORG'	03660000
	PUNCH '**'	03670000
&N	SETA &N+1	03680000
	AIF (&N LE &MAX) .EXECS	03690000
	PUNCH '          EJECT'	03700000
	EJECT	03710000
*		03720000
	PUNCH '**'	03730000
	PUNCH '** EXEC IMMEDIATE STATEMENT'	03740000
	PUNCH '**'	03750000
	PUNCH 'EXECIMMD SQLENTER'	03760000
	PUNCH '      EXEC SQL EXECUTE IMMEDIATE :STMT'	03770000
	PUNCH '          SQLRET'	03780000
	PUNCH '**'	03790000
	PUNCH '          LTORG'	03800000
	PUNCH '          EJECT'	03810000
	SPACE 3	03820000
	PUNCH '**'	03830000
	PUNCH '** COMMIT WORK STATEMENT'	03840000
	PUNCH '**'	03850000
	PUNCH 'COMMIT SQLENTER'	03860000
	PUNCH '      EXEC SQL COMMIT WORK'	03870000
	PUNCH '          SQLRET'	03880000
	PUNCH '**'	03890000
	PUNCH '          LTORG'	03900000
	PUNCH '          EJECT'	03910000
	SPACE 3	A019 03920000
	PUNCH '**'	A019 03930000
	PUNCH '** PREPARAE INTO STATEMENT'	A019 03940000
	PUNCH '**'	A019 03950000
	PUNCH 'PREPINTO SQLENTER DA'	A019 03960000
	PUNCH '      EXEC SQL PREPARE SX INTO :DA USING BOTH FROM :STMT'	A019 03970000
	PUNCH '          SQLRET'	A019 03980000
	PUNCH '**'	A019 03990000
	PUNCH '          LTORG'	A019 04000000
	PUNCH '          EJECT'	A019 04010000
&N	SETA 1	04020000
.DA	ANOP	04030000
	PUNCH 'SQLDA&N DSECT'	04040000
&N	SETA &N+1	04050000
	AIF (&N LE &MAX) .DA	04060000
	PUNCH 'DA          DSECT'	A019 04070000
	SPACE 3	04080000
	PUNCH 'MARKSQL CSECT'	SQL2 04090000
	PUNCH '      EXEC SQL BEGIN DECLARE SECTION'	SQL2 04100000
	PUNCH '**'	04110000
	PUNCH '** EXEC INCLUDE SQLDA STATEMENT'	04120000
	PUNCH '**'	04130000
	PUNCH '      EXEC SQL INCLUDE SQLDA'	04140000
	PUNCH '          SPACE 3'	04150000
	PUNCH '**'	04160000
	PUNCH '** DATA AREAS USED BY SQL STATEMENTS'	04170000
	PUNCH '**'	04180000
	PUNCH '          SPACE 3'	SQL2 04190000
	PUNCH 'STMTBUF DSECT'	04200000
	PUNCH 'STMT      DS      H,CL80'	04210000
	PUNCH '**'	04220000
	PUNCH 'SQLCODEX DSECT'	SQL2 04230000
	PUNCH 'SQLCODE DS      F'	SQL2 04240000
	PUNCH '**'	SQL2 04250000
	PUNCH 'DUMMY      DSECT'	SQL2 04260000
	PUNCH '**'	04270000
	PUNCH '      EXEC SQL END DECLARE SECTION'	SQL2 04280000
	PUNCH '**'	04290000
*	PUNCH '          END'	04300000
		04310000
	END	04320000

## Query Language Parameters – BQLPARM

### BQLPARM

```

QPOLBQL  TITLE 'BQLPARM - COMPUTER ASSOCIATES INTERNATIONAL, INC.'      00010000
          ISEQ  73,80                                                    00020000
*****
*                                                                 * 00030000
*                                                                 * 00040000
*          PROPRIETARY AND CONFIDENTIAL INFORMATION OF          * 00050000
*          COMPUTER ASSOCIATES INTERNATIONAL, INC.              * 00060000
*          USE RESTRICTED BY WRITTEN LICENSE AGREEMENT          * 00070000
*                                                                 * 00080000
*          DO NOT REMOVE THIS NOTICE                            * 00090000
*                                                                 * 00100000
*                                                                 * 00110000
*          COPYRIGHT (C) COMPUTER ASSOCIATES INTERNATIONAL, INC. * 00120000
*          AS AN UNPUBLISHED WORK. ALL RIGHTS RESERVED.         * 00130000
*                                                                 * 00140000
*****
          SPACE 3
BQLPARM  CSECT
*****
          QLPARM *****
*
* THIS CSECT IS USED TO DEFINE QL INSTALLATION DEPENDENT PARAMETERS.
*
*****
          DC      CL8'BQLPARM'
*
*****
* THIS TABLE GIVES THE VALUES FOR PAGE WIDTH, PAGE HEIGHT,
* DATE TYPE, DATE POSITION,
* COLUMN HEADING TYPE, PAGE NUMBER POSITION AND LABELS ON
* SUMMARY LINES FOR EACH TERMINAL TYPE TO BE REFERENCED IN THE
* 'PRINT CON' COMMAND. A BLANK ENTRY FOR PAGE WIDTH OR PAGE
* HEIGHT INDICATES THAT THE CORRESPONDING INSTALLATION DEFAULT
* FOR WIDTH OR HEIGHT IS TO BE USED.
* IN BATCH INSTALLATIONS THIS TABLE CAN BE USED TO DEFINE SPECIAL
* INSTALLATION DEFINED PAGE FORMATS. IF NO ENTRIES ARE PUT IN THIS
* TABLE, THEN THE 'PRINT CON' COMMAND SHOULD NOT BE USED.
*
* EACH ENTRY IN CONTAB IS 17 BYTES LONG AS SHOWN:
* BYTES 1 TO 4-ALPHANUMERIC TERMINAL TYPE DESIGNATOR
* BYTES 5 TO 7-PAGE WIDTH. BLANK,A TO E(LEFT JUSTIFIED) OR
* 1 TO 132(RIGHT JUSTIFIED).
* BYTES 8 TO 10-PAGE HEIGHT.SAME FORMAT AS PAGE WIDTH ENTRY.
* SEE REFERENCE MANUAL FOR PAGE SIZES
* CORRESPONDING TO LETTERS A THRU E.
* BYTE 11 -COLUMN HEADING TYPE-BLANK,F OR X.
* BYTES 12 TO 13-PAGE NUMBER POSITION-ONE OF BLANK,NP,LL,LR,MT,MB,UR.
* BYTE 14 -LABELS ON SUMMARY LINES-BLANK,L OR X.
* BYTES 15 TO 16-DATE POSN. ONE OF UL,UR,LL,LR,MT,MB,ND.
* -USE ND IF NO DATE REQUIRED.
* BYTE 17 -DATE TYPE: T FOR TODAY(DD/MM/YY,MM/DD/YY),
* D FOR DATE(JAN 1,1974) OR
* BLANK
*****
CONTABS  DS      0F
          DC      CL4'2741'          IBM 2741
          DC      CL3'132'           PAGE WIDTH
          DC      CL3' 66'           PAGE HEIGHT
          DC      CL7' UR UL '
          DC      CL4'TTY '          TELETYPE
          DC      CL3' 72'
          DC      CL3' 66'
          DC      CL7'XNP ULD'
          DC      CL4'VCOM'          BELL VUCOM

```

**BQLPARM (cont.)**

```

          DC      CL3' 72'                                00670000
          DC      CL3' 16'                                00680000
          DC      CL7'XNP ND '                            00690000
          DC      CL4'4013'                                00700000
          DC      CL3' 72'                                00710000
          DC      CL3' 35'                                00720000
          DC      C'XNP ND '                              00730000
CONTABE  EQU      *                                MUST BE AT END OF TABLE 00740000
          EJECT                                           00750000
*****
*                                           00760000
*                                           00770000
* MISCELLANEOUS QL PARAMETERS                                00780000
*                                           00790000
*****
*                                           00800000
*                                           00810000
CONTC    EQU      C'?'                                CHARACTER TO BE USED FOR LINE
*                                           CONTINUATION. ANY CHARACTER EXCEPT
*                                           ; MAY BE USED
*                                           00830000
*                                           00840000
*                                           00850000
ECHO     EQU      1                                SET TO 1 IF INPUT LINES ARE TO BE ECHO
*                                           PRINTED(OS/390 BATCH INSTALLATIONS.
*                                           SET TO 0 IF INPUT IS NOT TO BE ECHO
*                                           PRINTED(ONLINE VERSIONS).
*                                           00860000
*                                           00870000
*                                           00880000
*                                           00890000
*                                           00900000
TERM     EQU      1                                ERROR SEVERITY LEVEL THRESHOLD.
*                                           ALL ERROR MESSAGES
*                                           WITH MESSAGE SEVERITY LEVEL
*                                           GREATER THAN THIS NUMBER WILL BE
*                                           PRINTED AT THE TERMINAL. THE USER CAN
*                                           RESET THE DEFAULT VALUE GIVEN HERE BY
*                                           USING THE OPTIONS COMMAND.
*                                           THIS ITEM IS IGNORED IN THE
*                                           BATCH VERSIONS.
*                                           00910000
*                                           00920000
*                                           00930000
*                                           00940000
*                                           00950000
*                                           00960000
*                                           00970000
*                                           00980000
*                                           00990000
*                                           01000000
OPSYSD   EQU      C'BQL'                                3-CHARACTER OPERATING SYSTEM DESIGNATOR.
*                                           MUST BE ONE OF:CMS OQL BQL .
*                                           01010000
*                                           01020000
*                                           01030000
FDLN     EQU      8                                FILE DESIGNATOR LENGTH. SET AS FOLLOWS:
*                                           CMS-20
*                                           BQL-8
*                                           OQL-8
*                                           01040000
*                                           01050000
*                                           01060000
*                                           01070000
*                                           01080000
TLU      EQU      1                                SET TO 1 IF TABLE LOOKUP CAPABILITY
*                                           IS INSTALLED;0 OTHERWISE.
*                                           01090000
*                                           01100000
*                                           01110000
*****
          EJECT                                           01120000
*****
*                                           01130000
*                                           01140000
*                                           01150000
* THIS AREA HOLDS THE DEFAULT VALUES FOR THE FILE DESIGNATORS USED
* BY QL ITSELF AND FOR THOSE FILES WHICH MAY BE SPECIFIED BY THE
* USE COMMAND. THE CONTENTS OF EACH ENTRY ARE DEPENDENT ON THE
* OPERATING SYSTEM BEING USED AS FOLLOWS.
*                                           01160000
*                                           01170000
*                                           01180000
*                                           01190000
*                                           01200000
* FOR VM/CMS
*                                           01210000
*                                           01220000
* BYTES 1 TO 8-BLANK EXCEPT FOR $$ILIST WHICH SHOULD HAVE BYTES 1-4
* EITHER ' CON' OR ' PRT' ACCORDINGLY AS PRINT CON
* OR PRINT PRT IS TO BE ASSUMED AS THE DEFAULT
* WHEN NO PRINT COMMAND IS ENTERED. BYTES 5-8 OF
* $$ILIST SHOULD BE SET TO THE DEFAULT CONSOLE
* TYPE TO BE USED WHEN 'PRINT CON' WITH NO CONSOLE TYPE
* IS ENTERED. THIS CONSOLE TYPE MUST MATCH AN ENTRY IN
* THE CONTABS TABLE ABOVE.
*                                           01230000
*                                           01240000
*                                           01250000
*                                           01260000
*                                           01270000
*                                           01280000
*                                           01290000
*                                           01300000
*                                           01310000
* BYTES 9 TO 16-DEFAULT CMS FILETYPE NORMALLY THIS CORRESPONDS TO
* THE STANDARD DDNAME FOR THE FILE
*                                           01320000
*                                           01330000
*                                           01340000
* BYTES 17 TO 20-FILE MODE FOLLOWED BY 2 BLANKS, NORMALLY
* THIS WOULD BE 'P1 ' FOR CP AND 'A1 'FOR VM.
*                                           01350000
*                                           01360000
*                                           01370000

```

**BQLPARM (cont.)**

```

***** 01380000
* 01390000
* FOR VS1 AND VS2 BATCH AND TSO 01400000
* 01410000
* BYTES 1 TO 8-BLANK EXCEPT FOR $$ILIST WHICH SHOULD HAVE 01420000
* BYTES 1-4 SET TO 'PRT' NOTE THAT THE 01430000
* PRINT CON COMMAND WITH A CONSOLE TYPE MAY 01440000
* BE USED TO OBTAIN INSTALLATION DEFINED 01450000
* PAGE FORMATS SINCE CON IS CONSIDERED THE SAME 01460000
* AS PRT FOR BATCH. IF ENTRIES HAVE BEEN PLACED IN 01470000
* CONTAB, THEN BYTES 5 TO 8 OF $$ILIST 01480000
* SHOULD BE SET TO THE DEFAULT PAGE FORMAT TO BE USED 01490000
* WHEN 'PRINT CON' WITH NO CONSOLE TYPE IS USED. THIS 01500000
* CONSOLE TYPE MUST MATCH AN ENTRY IN CONTAB. IF NO 01510000
* ENTRIES HAVE BEEN PUT IN CONTAB, THEN BYTES 5 TO 8 01520000
* SHOULD BE BLANK. 01530000
* 01540000
***** 01550000
DS OF 01560000
$$IOLD DC CL(FDLN)' ' 01570000
$$INEW DC CL(FDLN)' ' 01580000
$$ITRAN DC CL(FDLN)' ' 01590000
$$ICRD1 DC CL(FDLN)' ' 01600000
$$ICRD2 DC CL(FDLN)' ' 01610000
$$ICRD3 DC CL(FDLN)' ' 01620000
$$ISBF1 DC CL(FDLN)' ' 01630000
$$ISBF2 DC CL(FDLN)' ' 01640000
$$ISBF3 DC CL(FDLN)' ' 01650000
$$ISBF4 DC CL(FDLN)' ' 01660000
$$ISBF5 DC CL(FDLN)' ' 01670000
$$ISLIB DC CL(FDLN)' ' 01680000
$$ILIB DC CL(FDLN)' ' 01690000
$$ISBF0 DC CL(FDLN)' ' 01700000
$$ILIST DC CL(FDLN)' CON2741' 01710000
* END OF USER DEFINED FILE DESIGNATOR DEFAULTS 01720000
EJECT 01730000
***** 01740000
* FILE DESIGNATORS FOR QL FILES. THE CONTENTS OF EACH DESIGNATOR 01750000
* DEPEND ON THE OPERATING SYSTEM USED AS FOLLOWS. 01760000
* 01770000
* FOR CP-67/CMS AND VM/370/CMS 01780000
* 01790000
* BYTES 1 TO 8-ANY VALID CMS FILENAME 01800000
* BYTES 9 TO 16-ANY VALID CMS FILETYPE 01810000
* BYTES 17 TO 20-FILE MODE FOLLOWED BY 2 BLANKS. 01820000
* 01830000
***** 01840000
* FOR VS1 AND VS2 BATCH AND TSO 01850000
* 01860000
* BYTES 1 TO 8-MUST CONTAIN DDNAME OF FILE 01870000
* 01880000
***** 01890000
SPACE 01900000
* QUERY FILE. HOLDS SOURCE CODE GENERATED BY QL. 01910000
$$QUERY DC CL(FDLN)'$QUERY' 01920000
SPACE 01930000
* SRC FILE. HOLDS QL SOURCE CODE AFTER RETAIN OR TERMINATION. 01940000
$$SRC DC CL(FDLN)'$SOURCE' 01950000
SPACE 01960000
* TEMP FILE. SCRATCH FILE HOLDING $QUERY CARD IMAGES. 01970000
$$TEMP DC CL(FDLN)'$TEMP' 01980000
SPACE 01990000
* TEMPE FILE. SCRATCH FILE FOR QL SOURCE AFTER EDIT CALL. 02000000
$$TEMPE DC CL(FDLN)'$TEMPE' 02010000
SPACE 02020000
* FILE HOLDING AUTOMATICALLY GENERATED FILE DEFINITION 02030000
$$SVFD DC CL(FDLN)'$SAVEFD' 02040000
SPACE 02050000
* MESSAGE FILE. HOLDS QL SIGNON MESSAGE. 02060000
MESSAGE DC CL(FDLN)'$MESSAGE' 02070000
SPACE 02080000

```

**BQLPARM (cont.)**

```

* DSN OF FILE CONTAINING MARKINIT MODULES. TSX ONLY                                02090000
M4CALL1 DC CL(FDLN) 'SYS1.ASMK4.MARKIV.VERSION3(MARKINIT) '                      02100000
      SPACE                                                                    02110000
* DSN OF FILE CONTAINING PROCESSING MODULES. TSX ONLY                                02120000
M4CALL2 DC CL(FDLN) 'SYS1.ASMK4.MARKIV.VERSION3(MARKIV) '                      02130000
      SPACE                                                                    02140000
* FILE CONTAINING GLOSSARY M4INPUT CARDS.                                          02150000
GLOSSF  DC CL(FDLN) '$GLOSS'                                                  02160000
      SPACE                                                                    02170000
* FILE CONTAINING RC CARD FOR REPORT PHASE. CMS,TSX ONLY.                        02180000
REPRC   DC CL(FDLN) ' '                                                       02190000
* END OF QL FILE DESIGNATORS                                                    02200000
      EJECT                                                                    02210000
*****                                                                            02220000
*                                                                                02230000
* QL MESSAGES                                                                    02240000
*                                                                                02250000
* THE WORDING, LENGTH OR LANGUAGE OF THESE MESSAGES MAY BE                    02260000
* MODIFIED TO SUIT A PARTICULAR INSTALLATION OR OPERATING SYSTEM.              02270000
* THE MAXIMUM LENGTH OF ANY MESSAGE IS 68 CHARACTERS.                          02280000
* THE WORDING OF THE FOLLOWING ERROR MESSAGES VARIES BETWEEN                    02290000
* OPERATING SYSTEMS: 9,12,19,26 AND 50.                                         02300000
*                                                                                02310000
*****                                                                            02320000
*                                                                                02330000
ERR1     DC C'SYNTAX ERROR'                                                    02340000
ERR2     DC C'MORE THAN 30 FIELDS IN LIST OR SAVE ST'                          02350000
ERR3     DC C'MORE THAN 1 SORT OR BREAK SUB-ST/LIST ST'                       02360000
ERR4     DC C'FILE DEFINITION NAME (FNAME OR TFNAME CMD) NOT GIVEN'            02370000
ERR5     DC C'OLD,NEW OR TRAN FILE(USE CMD) NOT GIVEN'                        02380000
ERR6     DC C'SFNAME SPECIFIED WITHOUT SLIB OR VICE VERSA'                    02390000
ERR7     DC C'CORD FILE QUALIFIER USED AND CFNAME CMD NOT ENTERED'            02400000
ERR8     DC C'LIB FILE( USE ST) NOT GIVEN'                                     02410000
ERR9     DC C'PROGRAM CONTAINS SERIOUS ERRORS.RECOMPILE'                      02420000
ERR10    DC C'NO PRECEDING LIST STATEMENT'                                     02430000
ERR11    DC C'UNDEFINED LABEL'                                                 02440000
ERR12    DC C'NO DDCARD PROVIDED FOR RETAIN FILE'                             02450000
ERR13    DC C'TYPE NO OR HIT CARRIAGE RETURN'                                02460000
ERR14    DC C'LENGTH OF NAME EXCEEDS 8 CHARACTERS'                           02470000
ERR15    DC C'WARNING.CONTROL BREAK OCCURS WITHOUT SORT.PRIOR LIST ST.'        02480000
ERR16    DC C'TOO MANY NESTED COPY STATEMENTS'                                02490000
ERR17    DC C'DUPLICATE LABELS'                                                02500000
ERR18    DC C'ILLEGAL BACKWARD BRANCH'                                        02510000
ERR19    DC C'PROGRAM CONTAINS ERRORS'                                         02520000
ERR20    DC C'TOO MANY LABELED STATEMENTS'                                     02530000
ERR21    DC C'CFNAME ENTERED AND -USE CORDN- NOT GIVEN'                       02540000
ERR22    DC C'TOO MANY PR OR TF CARDS GENERATED'                             02550000
ERR23    DC C'TOO MANY R1 CARDS GENERATED'                                    02560000
ERR24    DC C'WARNING.PRINT CMD(IF USED) SHOULD PRECEDE ALL LIST ST.'          02570000
ERR25    DC C'EQUATE CMD(IF USED) MUST PRECEDE ALL STMNTS'                    02580000
ERR26    DC C'NO DDCARD PROVIDED FOR COPY FILE'                              02590000
ERR27    DC C'LENGTH OF CHARACTER LITERAL >31'                               02600000
ERR28    DC C'COMMA SEPARATOR MAY ONLY FOLLOW EQ OR NE'                       02610000
ERR29    DC C'MORE THAN 9 SORT, BREAK, OR SUMMARY ITEMS GIVEN'                02620000
ERR30    DC C'ITEM NO. 0 OR > NO. OF ITEMS IN LIST OR SAVE STATEMENT'          02630000
ERR31    DC C'BREAK SUB-ST MUST PRECEDE NEWPAGE,SUBTITLE OR SUMMARIES'        02640000
ERR32    DC C'AUTOCORD OR CFNAME CMDS(IF USED) MUST PRECEDE ALL STMNTS'       02650000
ERR33    DC C'SFORMAT CMD(IF USED) MUST PRECEDE ALL SAVE/SAVEALL ST'          02660000
ERR34    DC C'TITLE IS LONGER THAN 59 CHARS'                                   02670000
ERR35    DC C'MORE THAN 3 TITLE LINES SPECIFIED'                             02680000
ERR36    DC C'EDIT STRING IN LAYOUT PH CONTAINS>15 CHARS'                    02690000
ERR37    DC C'WARNING. MORE ITEMS IN LAYOUT PH THAN LIST PH'                  02700000
ERR38    DC C'NO. OF SPACES SPECIFIED IN LAYOUT PH IS >99.'                   02710000
ERR39    DC C'PAGE WIDTH OR HEIGHT=0 OR > 132.ITEM IGNORED'                  02720000
ERR40    DC C'SP OR I VALUE=0 OR >9.ITEM IGNORED'                             02730000
ERR41    DC C'PAGEN OR MAXP VALUE=0 OR >9999.ITEM IGNORED'                   02740000
ERR42    DC C'MAXL VALUE=0 OR >99.ITEM IGNORED'                               02750000
ERR43    DC C'SUMMARY SPECIFIED ON CONTROL BREAK FIELD.PRIOR LIST ST.'        02760000
ERR44    DC C'KEY SUB-ST AND NO SAVE ST. WITH SFNAME PHRASE GIVEN'            02770000
ERR45    DC C'HEADING CONTAINS MORE THAN 14 CHARACTERS'                      02780000
ERR46    DC C'INITIAL VALUE CONTAINS >16 CHARACTERS'                         02790000

```

**BQLPARM (cont.)**

ERR47	DC	C'DEFINE EDIT STRING CONTAINS >3 CHARACTERS'	02800000
ERR48	DC	C'TWO FILE DESIGNATORS GIVEN AND CFNAME NOT ENTERED'	02810000
ERR49	DC	C'WARNING. FILE DOES NOT EXIST. FN,FT,AND MODE ARE:'	02820000
ERR50	DC	C'EDIT NOT SUPPORTED IN THIS VERSION'	02830000
ERR51	DC	C'NO STATEMENTS ENTERED AND ONLY TRAN OR OLD GIVEN'	02840000
ERR52	DC	C'END COMMAND NOT ENTERED'	02850000
ERR53	DC	C'SYNONYM TABLE FULL'	02860000
ERR54	DC	C'LINE CONTAINS MORE THAN 64 CHARACTERS'	02870000
ERR55	DC	C'ONLY START,QUIT,EDIT,RETAIN,OR CLEAR ALLOWED AFTER END'	02880000
ERR56	DC	C'MORE THAN 1 WHERE SUB-ST PER LIST OR SAVE ST'	02890000
ERR57	DC	C'NOT ENOUGH CORE FOR MARK IV. MINIMUM OF 384K REQUIRED'	02900000
ERR58	DC	C'LET EXPRESSION TOO LONG'	02910000
ERR59	DC	C'MORE THAN 9 DEC PLACES SPECIFIED'	02920000
ERR60	DC	C'FIELD LENGTH=0 OR > MAXIMUM ALLOWED LENGTH'	02930000
ERR61	DC	C'INVALID CONSOLE DESIGNATOR. CMD IGNORED'	02940000
ERR62	DC	C'TOO MANY CONTINUATION CARDS. JOB TERMINATED'	02950000
ERR63	DC	C'WARNING.NEXT LINE MUST NOT BE CONTINUED'	02960000
ERR64	DC	C'FILE DESIGNATOR TOO LONG'	02970000
ERR65	DC	C'PARTIAL FIELD START OR LENGTH=0 OR > 99'	02980000
ERR66	DC	C'SOURCE FILE DOES NOT EXIST. CMD IGNORED'	02990000
ERR67	DC	C'APOSTROPHE MISSING IN STRING'	03000000
ERR68	DC	C'WARNING.MORE THAN 1 REFERENCE TO SAME ITEM IN SUB-ST.'	03010000
ERR69	DC	C'CONTROL BREAK LEVEL=0 OR >9'	03020000
ERR70	DC	C'WARNING.BREAK(IF USED)MUST ASSIGN ALL CONTROL BRK LEVELS'	03030000
ERR71	DC	C'PARENTHESIS NESTED TO DEPTH > 9'	03040000
ERR72	DC	C'SAME NUMERATOR USED MORE THAN ONCE IN PERCENT/RATIO'	03050000
ERR73	DC	C'CHARACTER STRING SPLIT OVER 2 LINES'	03060000
ERR74	DC	C'CHARACTER STRING APPEARS IN ARITHMETIC EXPRESSION'	03070000
ERR75	DC	C'EDIT STRING GIVEN WITH C,V,OR E TYPE FIELDS'	03080000
ERR76	DC	C'DUPLICATE FILE ID IN SAVE STATEMENT'	03090000
ERR77	DC	C'MORE THAN 1 KEY SUB-ST. PER SAVE ST.'	03100000
ERR78	DC	C'TEMPORARY FIELD IS ALREADY DEFINED'	03110000
ERR79	DC	C'COPY OR EDIT COMMAND MUST BE LAST SENTENCE IN LINE'	03120000
ERR80	DC	C'INVALID DECIMAL NUMBER'	03130000
ERR81	DC	C'UNKNOWN KEYWORD. KEYWORD-VALUE PAIR IGNORED'	03140000
ERR82	DC	C'ILLEGAL VALUE FOR ITEM.KEYWORD-VALUE PAIR IGNORED'	03150000
ERR83	DC	C'BLOCKING FACTOR/BUFFER SIZE=0 OR > 9999'	03160000
ERR84	DC	C'ONLY 1 SAVE ST. PER SUBFILE MAY USE SFNAME'	03170000
ERR85	DC	C'MORE THAN 5 SUBFILE SELECTORS IN SAVEALL ST.'	03180000
ERR86	DC	C'NO PRECEDING LIST,SAVE OR SAVEALL ST.'	03190000
ERR87	DC	C'NO. OF SUBFILES SPECIFIED NE NO. OF FILE SELECTORS'	03200000
ERR88	DC	C'USE CMD NOT ENTERED FOR:'	03210000
ERR89	DC	C'TABLE LOOKUP FEATURE NOT SUPPORTED'	03220000
ERR90	DC	C'OVERRIDE OPERAND NOT SPECIFIED.'	OVER 03230000
ERRN	EQU	*	03240000
	EJECT		03250000
*****			03260000
*			03270000
**** DO NOT MAKE CHANGES BEYOND HERE ****			03280000
*			03290000
*****			03300000
* PRINT OFF			03310000
CONTAB	DC	A (CONTABS,17,CONTABE)	03320000
*			03330000
	DS	OF	03340000
\$IECHO	DC	A (ECHO)	03350000
ITERM	DC	A (TERM)	03360000
CONTCH	DC	AL1 (CONTC)	03370000
*			03380000
QLMMSG	DC	A (MSGADR1-3, (MSGADR1-MSGADR1) /3)	03390000
MSGADR1	DC	AL3 (ERR1)	03400000
	DC	AL3 (ERR2)	03410000
	DC	AL3 (ERR3)	03420000
	DC	AL3 (ERR4)	03430000
	DC	AL3 (ERR5)	03440000
	DC	AL3 (ERR6)	03450000
	DC	AL3 (ERR7)	03460000
	DC	AL3 (ERR8)	03470000
	DC	AL3 (ERR9)	03480000
	DC	AL3 (ERR10)	03490000
	DC	AL3 (ERR11)	03500000

**BQLPARM (cont.)**

DC	AL3 (ERR12)	03510000
DC	AL3 (ERR13)	03520000
DC	AL3 (ERR14)	03530000
DC	AL3 (ERR15)	03540000
DC	AL3 (ERR16)	03550000
DC	AL3 (ERR17)	03560000
DC	AL3 (ERR18)	03570000
DC	AL3 (ERR19)	03580000
DC	AL3 (ERR20)	03590000
DC	AL3 (ERR21)	03600000
DC	AL3 (ERR22)	03610000
DC	AL3 (ERR23)	03620000
DC	AL3 (ERR24)	03630000
DC	AL3 (ERR25)	03640000
DC	AL3 (ERR26)	03650000
DC	AL3 (ERR27)	03660000
DC	AL3 (ERR28)	03670000
DC	AL3 (ERR29)	03680000
DC	AL3 (ERR30)	03690000
DC	AL3 (ERR31)	03700000
DC	AL3 (ERR32)	03710000
DC	AL3 (ERR33)	03720000
DC	AL3 (ERR34)	03730000
DC	AL3 (ERR35)	03740000
DC	AL3 (ERR36)	03750000
DC	AL3 (ERR37)	03760000
DC	AL3 (ERR38)	03770000
DC	AL3 (ERR39)	03780000
DC	AL3 (ERR40)	03790000
DC	AL3 (ERR41)	03800000
DC	AL3 (ERR42)	03810000
DC	AL3 (ERR43)	03820000
DC	AL3 (ERR44)	03830000
DC	AL3 (ERR45)	03840000
DC	AL3 (ERR46)	03850000
DC	AL3 (ERR47)	03860000
DC	AL3 (ERR48)	03870000
DC	AL3 (ERR49)	03880000
DC	AL3 (ERR50)	03890000
DC	AL3 (ERR51)	03900000
DC	AL3 (ERR52)	03910000
DC	AL3 (ERR53)	03920000
DC	AL3 (ERR54)	03930000
DC	AL3 (ERR55)	03940000
DC	AL3 (ERR56)	03950000
DC	AL3 (ERR57)	03960000
DC	AL3 (ERR58)	03970000
DC	AL3 (ERR59)	03980000
DC	AL3 (ERR60)	03990000
DC	AL3 (ERR61)	04000000
DC	AL3 (ERR62)	04010000
DC	AL3 (ERR63)	04020000
DC	AL3 (ERR64)	04030000
DC	AL3 (ERR65)	04040000
DC	AL3 (ERR66)	04050000
DC	AL3 (ERR67)	04060000
DC	AL3 (ERR68)	04070000
DC	AL3 (ERR69)	04080000
DC	AL3 (ERR70)	04090000
DC	AL3 (ERR71)	04100000
DC	AL3 (ERR72)	04110000
DC	AL3 (ERR73)	04120000
DC	AL3 (ERR74)	04130000
DC	AL3 (ERR75)	04140000
DC	AL3 (ERR76)	04150000
DC	AL3 (ERR77)	04160000
DC	AL3 (ERR78)	04170000
DC	AL3 (ERR79)	04180000
DC	AL3 (ERR80)	04190000
DC	AL3 (ERR81)	04200000
DC	AL3 (ERR82)	04210000

**BQLPARM (cont.)**

DC	AL3 (ERR83)	04220000
DC	AL3 (ERR84)	04230000
DC	AL3 (ERR85)	04240000
DC	AL3 (ERR86)	04250000
DC	AL3 (ERR87)	04260000
DC	AL3 (ERR88)	04270000
DC	AL3 (ERR89)	04280000
DC	AL3 (ERR90)	04290000
MSGADRN	DC AL3 (ERRN)	OVER 04300000
EJECT		04310000
*****		04320000
* FILE DESIGNATOR AREA. HOLDS CURRENT VALUE OF ALL THE FILE		04330000
* DESIGNATORS WHICH CAN BE REFERENCED FROM QL. THE DEFAULT VALUES OF		04340000
* THESE DESIGNATORS ARE GIVEN ABOVE.		04350000
* THE LAYOUT OF EACH DESIGNATOR IS OPERATING SYSTEM DEPENDENT.		04360000
*		04370000
*****		04380000
FDTAB	DC A (\$\$OLD,FDLN,LASTFD) USED TO COPY DEFAULTS FROM	04390000
*	AREA ABOVE. DONE BY REMOTE 4.	04400000
\$FDLEN	DC A (FDLN)	04410000
\$\$OLD	DC (FDLN) C ' '	04420000
\$\$NEW	DC (FDLN) C ' '	04430000
\$\$TRAN	DC (FDLN) C ' '	04440000
\$\$CORD1	DC (FDLN) C ' '	04450000
\$\$CORD2	DC (FDLN) C ' '	04460000
\$\$CORD3	DC (FDLN) C ' '	04470000
\$\$SUBF1	DC (FDLN) C ' '	04480000
\$\$SUBF2	DC (FDLN) C ' '	04490000
\$\$SUBF3	DC (FDLN) C ' '	04500000
\$\$SUBF4	DC (FDLN) C ' '	04510000
\$\$SUBF5	DC (FDLN) C ' '	04520000
\$\$SLIB	DC (FDLN) C ' '	04530000
\$\$LIB	DC (FDLN) C ' '	04540000
\$\$SUBF0	DC (FDLN) C ' '	04550000
\$\$LIST	DC (FDLN) C ' '	04560000
LASTFD	EQU *-FDLN MUST FOLLOW LAST FILE DESIGNATOR	04570000
* END OF FILE DESIGNATOR AREA		04580000
EJECT		04590000
*****		04600000
* MISCELLANEOUS FLAGS. THE INITIAL VALUES OF \$TERM AND		04610000
* \$ECHO ARE COPIED FROM ABOVE BY REMOTE 4.		04620000
* FNAME,GLOSS,\$PRORUN AND \$RC ARE INITIALIZED DIRECTLY		04630000
* BY REMOTE 4.		04640000
*		04650000
*****		04660000
*		04670000
\$\$FNAME	DS CL8 FILENAME (FOR RC CARD) .	04680000
*	USED FOR GLOSSARY RUN	04690000
\$\$GLOSS	DS CL4 1ST BYTE=A,B,OR 1.IS PUT ON FD	04700000
*	CARD WHEN PRODUCING A GLOSSARY	04710000
\$PRORUN	DS F =1 IF THIS IS A PROCESSING RUN	04720000
\$RC	DS F RETURN CODE	04730000
*		04740000
\$ECHO	DS F =1 IF SOURCE IS TO BE ECHO PRINTED	04750000
\$TERM	DS F FOR ONLINE SYSTEMS,GIVES THE MINIMUM	04760000
*	MESSAGE SEVERITY LEVEL FOR MESSAGES	04770000
*	PRINTED AT THE TERMINAL	04780000
OPSYS	DC AL3 (OPSYSD)	04790000
*		04800000
\$MK4SPF	DC A (TLU) TLU CAPABILITY FLAGS	04810000
ENTRY	CONTAB,QLMMSGA,\$IECHO,OPSYS,ITERM,CONTC	04820000
ENTRY	\$\$IOLD,\$\$INew,\$\$ITRAN,\$\$ICRD1,\$\$ICRD2,\$\$ICRD3	04830000
ENTRY	\$\$ISBF0,\$\$ISBF1,\$\$ISBF2,\$\$ISBF3,\$\$ISBF4,\$\$ISBF5	04840000
ENTRY	\$\$ISBF3,\$\$ISBF4,\$\$ISBF5	04850000
ENTRY	\$\$QUERY,\$\$SRC,\$\$TEMP,\$\$TEMPE,MESSAGE,\$\$SVFD	04860000
ENTRY	\$RC,\$TERM,\$PRORUN,\$ECHO,\$\$FNAME,\$\$GLOSS	04870000
ENTRY	\$\$CORD1,\$\$CORD2,\$\$CORD3,\$\$SUBF3,\$\$SUBF4,\$\$SUBF5	04880000
ENTRY	\$\$OLD,\$\$NEW,\$\$TRAN,\$\$SUBF0,\$\$SUBF1,\$\$SUBF2	04890000
ENTRY	\$\$LIST,\$\$LIB,\$\$SLIB,FDTAB,\$FDLEN,REPRC,M4CALL1,M4CALL2	04900000
ENTRY	GLOSSF,\$MK4SPF	04910000
END		04920000

# Online Language Parameters – OQLPARM

## OQLPARM

```

QPOLTSO  TITLE 'OQLPARM - COMPUTER ASSOCIATTES INTERNATIONAL, INC.'      00010000
        ISEQ  73,80                                                         00020000
*****
*                                                         * 00030000
*                                                         * 00040000
*           PROPRIETARY AND CONFIDENTIAL INFORMATION OF      * 00050000
*           COMPUTER ASSOCIATES INTERNATIONAL, INC.          * 00060000
*           USE RESTRICTED BY WRITTEN LICENSE AGREEMENT      * 00070000
*                                                         * 00080000
*           DO NOT REMOVE THIS NOTICE                        * 00090000
*                                                         * 00100000
*                                                         * 00110000
*           COPYRIGHT (C) COMPUTER ASSOCIATES INTERNATIONAL, * 00120000
*           AS AN UNPUBLISHED WORK. ALL RIGHTS RESERVED.     * 00130000
*                                                         * 00140000
*****                                                         00150000
        SPACE 3                                                            00160000
OQLPARM  CSECT                                                            00170000
***** QLPARM *****                                                    00180000
*                                                         00190000
* THIS CSECT IS USED TO DEFINE QL INSTALLATION DEPENDENT PARAMETERS. 00200000
*                                                         00210000
*****                                                         00220000
*                                                         00230000
*           DC      CL8'OQLPARM'                                         00240000
*                                                         00250000
*****                                                         00260000
*                                                         00270000
* THIS TABLE GIVES THE VALUES FOR PAGE WIDTH, PAGE HEIGHT,      00280000
* DATE TYPE, DATE POSITION,                                          00290000
* COLUMN HEADING TYPE, PAGE NUMBER POSITION AND LABELS ON          00300000
* SUMMARY LINES FOR EACH TERMINAL TYPE TO BE REFERENCED IN THE    00310000
* 'PRINT CON' COMMAND. A BLANK ENTRY FOR PAGE WIDTH OR PAGE        00320000
* HEIGHT INDICATES THAT THE CORRESPONDING INSTALLATION DEFAULT    00330000
* FOR WIDTH OR HEIGHT IS TO BE USED.                                00340000
*           IN BATCH INSTALLATIONS THIS TABLE CAN BE USED TO DEFINE SPECIAL
* INSTALLATION DEFINED PAGE FORMATS. IF NO ENTRIES ARE PUT IN THIS 00350000
* TABLE, THEN THE 'PRINT CON' COMMAND SHOULD NOT BE USED.        00360000
*                                                         00370000
*                                                         00380000
* EACH ENTRY IN CONTAB IS 17 BYTES LONG AS SHOWN:                 00390000
* BYTES 1 TO 4-ALPHANUMERIC TERMINAL TYPE DESIGNATOR              00400000
* BYTES 5 TO 7-PAGE WIDTH. BLANK,A TO E(LEFT JUSTIFIED) OR        00410000
* 1 TO 132(RIGHT JUSTIFIED).                                       00420000
* BYTES 8 TO 10-PAGE HEIGHT.SAME FORMAT AS PAGE WIDTH ENTRY.      00430000
*           SEE REFERENCE MANUAL FOR PAGE SIZES                    00440000
*           CORRESPONDING TO LETTERS A THRU E.                    00450000
* BYTE 11      -COLUMN HEADING TYPE-BLANK,F OR X.                 00460000
* BYTES 12 TO 13-PAGE NUMBER POSITION-ONE OF BLANK,NP,LL,LR,MT,MB.  00470000
* BYTE 14      -LABELS ON SUMMARY LINES-BLANK,L OR X.             00480000
* BYTES 15 TO 16-DATE POSN. ONE OF UL,UR,LL,LR,MT,MB,ND.          00490000
*           -USE ND IF NO DATE REQUIRED.                             00500000
* BYTE 17      -DATE TYPE: T FOR TODAY (DD/MM/YY,MM/DD/YY),       00510000
*           D FOR DATE(JAN 1,1974) OR                               00520000
*           BLANK                                                    00530000
*                                                         00540000
*****                                                         00550000
*                                                         00560000
CONTABS  DS      0F                                                         00570000
        DC      CL4'2741'          IBM 2741                             00580000
        DC      CL3'120'           PAGE WIDTH                         00590000
        DC      CL3' 32'           PAGE HEIGHT                        00600000
        DC      CL7'XNP ULD'                                               00610000
        DC      CL4'TTY '          TELETYPE                           00620000
        DC      CL3' 72'                                                   00630000
        DC      CL3' 66'                                                   00640000
        DC      CL7'XNP ULD'                                               00650000
        DC      CL4'VCOM'          BELL VUCOM                         00660000

```

**OQLPARM (cont.)**

```

          DC      CL3' 72'                                00670000
          DC      CL3' 16'                                00680000
          DC      CL7'XNP ND '                            00690000
          DC      CL4'4013'                                00700000
          DC      CL3' 72'                                00710000
          DC      CL3' 35'                                00720000
          DC      C'XNP ND '                              00730000
CONTABE  EQU      *                                     MUST BE AT END OF TABLE 00740000
          EJECT                                           00750000
*****
*
* MISCELLANEOUS QL PARAMETERS                            00780000
*
*****
*
CONTC    EQU      C'?'                                CHARACTER TO BE USED FOR LINE
*                                                CONTINUATION. ANY CHARACTER EXCEPT
*                                                ; MAY BE USED
*
ECHO     EQU      0                                SET TO 1 IF INPUT LINES ARE TO BE ECHO
*                                                PRINTED(O/S-360 BATCH INSTALLATIONS.
*                                                SET TO 0 IF INPUT IS NOT TO BE ECHO
*                                                PRINTED(ONLINE VERSIONS).
*
TERM     EQU      1                                ERROR SEVERITY LEVEL THRESHOLD.
*                                                ALL ERROR MESSAGES
*                                                WITH MESSAGE SEVERITY LEVEL
*                                                GREATER THAN THIS NUMBER WILL BE
*                                                PRINTED AT THE TERMINAL. THE USER CAN
*                                                RESET THE DEFAULT VALUE GIVEN HERE BY
*                                                USING THE OPTIONS COMMAND.
*                                                THIS ITEM IS IGNORED IN THE
*                                                BATCH VERSIONS.
*
OPSYSD   EQU      C'OQL'                            3-CHARACTER OPERATING SYSTEM DESIGNATOR.
*                                                MUST BE ONE OF:CMS OQL BQL .
*
FDLN     EQU      8                                FILE DESIGNATOR LENGTH. SET AS FOLLOWS:
*                                                CMS-20
*                                                BQL-8
*                                                OQL-8
*
TLU      EQU      1                                SET TO 1 IF TABLE LOOKUP CAPABILITY
*                                                IS INSTALLED;0 OTHERWISE.
*****
          EJECT
*****
* THIS AREA HOLDS THE DEFAULT VALUES FOR THE FILE DESIGNATORS USED
* BY QL ITSELF AND FOR THOSE FILES WHICH MAY BE SPECIFIED BY THE
* USE COMMAND. THE CONTENTS OF EACH ENTRY ARE DEPENDENT ON THE
* OPERATING SYSTEM BEING USED AS FOLLOWS.
*
* FOR CP-67/CMS AND VM370/CMS
*
* BYTES 1 TO 8-BLANK EXCEPT FOR $$ILIST WHICH SHOULD HAVE BYTES 1-4
* EITHER ' CON' OR ' PRT' ACCORDINGLY AS PRINT CON
* OR PRINT PRT IS TO BE ASSUMED AS THE DEFAULT
* WHEN NO PRINT COMMAND IS ENTERED. BYTES 5-8 OF
* $$ILIST SHOULD BE SET TO THE DEFAULT CONSOLE
* TYPE TO BE USED WHEN 'PRINT CON' WITH NO CONSOLE TYPE
* IS ENTERED. THIS CONSOLE TYPE MUST MATCH AN ENTRY IN
* THE CONTABS TABLE ABOVE.
*
* BYTES 9 TO 16-DEFAULT CMS FILETYPE NORMALLY THIS CORRESPONDS TO
* THE STANDARD DDNAME FOR THE FILE
*
* BYTES 17 TO 20-FILE MODE FOLLOWED BY 2 BLANKS, NORMALLY
* THIS WOULD BE 'P1 ' FOR CP AND 'A1 'FOR VM.
*

```

**OQLPARM (cont.)**

```

***** 01380000
* 01390000
* FOR VS1 AND VS2 BATCH AND TSO 01400000
* 01410000
* BYTES 1 TO 8-BLANK EXCEPT FOR $$ILIST WHICH SHOULD HAVE 01420000
* BYTES 1-4 SET TO 'PRT' NOTE THAT THE 01430000
* PRINT CON COMMAND WITH A CONSOLE TYPE MAY 01440000
* BE USED TO OBTAIN INSTALLATION DEFINED 01450000
* PAGE FORMATS SINCE CON IS CONSIDERED THE SAME 01460000
* AS PRT FOR BATCH. IF ENTRIES HAVE BEEN PLACED IN 01470000
* CONTAB, THEN BYTES 5 TO 8 OF $$ILIST 01480000
* SHOULD BE SET TO THE DEFAULT PAGE FORMAT TO BE USED 01490000
* WHEN 'PRINT CON' WITH NO CONSOLE TYPE IS USED. THIS 01500000
* CONSOLE TYPE MUST MATCH AN ENTRY IN CONTAB. IF NO 01510000
* ENTRIES HAVE BEEN PUT IN CONTAB, THEN BYTES 5 TO 8 01520000
* SHOULD BE BLANK. 01530000
* 01540000
***** 01550000
          DS      0F 01560000
$$IOLD DC CL(FDLN)' ' 01570000
$$INew DC CL(FDLN)' ' 01580000
$$ITRAN DC CL(FDLN)' ' 01590000
$$ICRD1 DC CL(FDLN)' ' 01600000
$$ICRD2 DC CL(FDLN)' ' 01610000
$$ICRD3 DC CL(FDLN)' ' 01620000
$$ISBF1 DC CL(FDLN)' ' 01630000
$$ISBF2 DC CL(FDLN)' ' 01640000
$$ISBF3 DC CL(FDLN)' ' 01650000
$$ISBF4 DC CL(FDLN)' ' 01660000
$$ISBF5 DC CL(FDLN)' ' 01670000
$$ISLIB DC CL(FDLN)' ' 01680000
$$ILIB DC CL(FDLN)' ' 01690000
$$ISBF0 DC CL(FDLN)' ' 01700000
$$ILIST DC CL(FDLN)' CON2741' 01710000
* END OF USER DEFINED FILE DESIGNATOR DEFAULTS 01720000
      EJECT 01730000
***** 01740000
* FILE DESIGNATORS FOR QL FILES. THE CONTENTS OF EACH DESIGNATOR 01750000
* DEPEND ON THE OPERATING SYSTEM USED AS FOLLOWS. 01760000
* 01770000
* FOR CP-67/CMS AND VM/370/CMS 01780000
* 01790000
* BYTES 1 TO 8-ANY VALID CMS FILENAME 01800000
* BYTES 9 TO 16-ANY VALID CMS FILETYPE 01810000
* BYTES 17 TO 20-FILE MODE FOLLOWED BY 2 BLANKS. 01820000
* 01830000
***** 01840000
* FOR VS1 AND VS2 BATCH AND TSO 01850000
* 01860000
* BYTES 1 TO 8-MUST CONTAIN DDNAME OF FILE 01870000
* 01880000
***** 01890000
      SPACE 01900000
* QUERY FILE. HOLDS SOURCE CODE GENERATED BY QL. 01910000
$$QUERY DC CL(FDLN)'$QUERY' 01920000
      SPACE 01930000
* SRC FILE. HOLDS QL SOURCE CODE AFTER RETAIN OR TERMINATION. 01940000
$$SRC DC CL(FDLN)'$SOURCE' 01950000
      SPACE 01960000
* TEMP FILE. SCRATCH FILE HOLDING $QUERY CARD IMAGES. 01970000
$$TEMP DC CL(FDLN)'$TEMP' 01980000
      SPACE 01990000
* TEMPE FILE. SCRATCH FILE FOR QL SOURCE AFTER EDIT CALL. 02000000
$$TEMPE DC CL(FDLN)'$TEMPE' 02010000
      SPACE 02020000
* FILE HOLDING AUTOMATICALLY GENERATED FILE DEFINITION 02030000
$$SVFD DC CL(FDLN)'$SAVEFD' 02040000
      SPACE 02050000
* MESSAGE FILE. HOLDS QL SIGNON MESSAGE. 02060000
MESSAGE DC CL(FDLN)'$MESSAGE' 02070000
      SPACE 02080000

```

**OQLPARM (cont.)**

* DSN OF FILE CONTAINING MARKINIT MODULES. TSX ONLY	02090000
M4CALL1 DC CL(FDLN)'SYS1.ASMK4.MARKIV.VERSION3(MARKINIT)'	02100000
SPACE	02110000
* DSN OF FILE CONTAINING PROCESSING MODULES. TSX ONLY	02120000
M4CALL2 DC CL(FDLN)'SYS1.ASMK4.MARKIV.VERSION3(MARKIV)'	02130000
SPACE	02140000
* FILE CONTAINING GLOSSARY M4INPUT CARDS.	02150000
GLOSSF DC CL(FDLN)'\$GLOSS'	02160000
SPACE	02170000
* FILE CONTAINING RC CARD FOR REPORT PHASE. CMS,TSX ONLY.	02180000
REPRC DC CL(FDLN)' '	02190000
* END OF QL FILE DESIGNATORS	02200000
EJECT	02210000
*****	02220000
*	02230000
* QL MESSAGES	02240000
*	02250000
* THE WORDING, LENGTH OR LANGUAGE OF THESE MESSAGES MAY BE	02260000
* MODIFIED TO SUIT A PARTICULAR INSTALLATION OR OPERATING SYSTEM.	02270000
* THE MAXIMUM LENGTH OF ANY MESSAGE IS 68 CHARACTERS.	02280000
* THE WORDING OF THE FOLLOWING ERROR MESSAGES VARIES BETWEEN	02290000
* OPERATING SYSTEMS: 9,12,19,26 AND 50.	02300000
*	02310000
*****	02320000
*	02330000
ERR1 DC C'SYNTAX ERROR'	02340000
ERR2 DC C'MORE THAN 30 FIELDS IN LIST OR SAVE ST'	02350000
ERR3 DC C'MORE THAN 1 SORT OR BREAK SUB-ST/LIST ST'	02360000
ERR4 DC C'FILE DEFINITION NAME (FNAME OR TFNAME CMD) NOT GIVEN'	02370000
ERR5 DC C'OLD,NEW OR TRAN FILE(USE CMD) NOT GIVEN'	02380000
ERR6 DC C'SFNAME SPECIFIED WITHOUT SLIB OR VICE VERSA'	02390000
ERR7 DC C'CORD FILE QUALIFIER USED AND CFNAME CMD NOT ENTERED'	02400000
ERR8 DC C'LIB FILE( USE ST) NOT GIVEN'	02410000
ERR9 DC C'PROGRAM CONTAINS ERRORS.CORRECT USING EDIT'	02420000
ERR10 DC C'NO PRECEDING LIST STATEMENT'	02430000
ERR11 DC C'UNDEFINED LABEL'	02440000
ERR12 DC C'NO DD CARD PROVIDED FOR RETAIN FILE'	02450000
ERR13 DC C'TYPE NO OR HIT CARRIAGE RETURN'	02460000
ERR14 DC C'LENGTH OF NAME EXCEEDS 8 CHARACTERS'	02470000
ERR15 DC C'WARNING.CONTROL BREAK OCCURS WITHOUT SORT.PRIOR LIST ST.'	02480000
ERR16 DC C'TOO MANY NESTED COPY STATEMENTS'	02490000
ERR17 DC C'DUPLICATE LABELS'	02500000
ERR18 DC C'ILLEGAL BACKWARD BRANCH'	02510000
ERR19 DC C'PROGRAM CONTAINS ERRORS.CORRECT AND REENTER END'	02520000
ERR20 DC C'TOO MANY LABELED STATEMENTS'	02530000
ERR21 DC C'CFNAME ENTERED AND -USE CORDN- NOT GIVEN'	02540000
ERR22 DC C'TOO MANY PR OR TF CARDS GENERATED'	02550000
ERR23 DC C'TOO MANY R1 CARDS GENERATED'	02560000
ERR24 DC C'WARNING.PRINT CMD(IF USED) SHOULD PRECEDE ALL LIST ST.'	02570000
ERR25 DC C'EQUATE CMD(IF USED) MUST PRECEDE ALL STMNTS'	02580000
ERR26 DC C'NO DD CARD PROVIDED FOR COPY FILE'	02590000
ERR27 DC C'LENGTH OF CHARACTER LITERAL >31'	02600000
ERR28 DC C'COMMA SEPARATOR MAY ONLY FOLLOW EQ OR NE'	02610000
ERR29 DC C'MORE THAN 9 SORT,BREAK OR SUMMARY ITEMS GIVEN'	02620000
ERR30 DC C'ITEM NO. 0 OR > NO. OF ITEMS IN LIST OR SAVE STATEMENT'	02630000
ERR31 DC C'BREAK SUB-ST MUST PRECEDE NEWPAGE,SUBTITLE,OR SUMMARIES'	02640000
ERR32 DC C'AUTOCORD OR CFNAME CMDS(IF USED) MUST PRECEDE ALL STMNTS'	02650000
ERR33 DC C'SFORMAT CMD(IF USED) MUST PRECEDE ALL SAVE/SAVEALL ST'	02660000
ERR34 DC C'TITLE IS LONGER THAN 59 CHARS'	02670000
ERR35 DC C'MORE THAN 3 TITLE LINES SPECIFIED'	02680000
ERR36 DC C'EDIT STRING IN LAYOUT PH CONTAINS>15 CHARS'	02690000
ERR37 DC C'WARNING. MORE ITEMS IN LAYOUT PH THAN LIST PH'	02700000
ERR38 DC C'NO. OF SPACES SPECIFIED IN LAYOUT PH IS >99.'	02710000
ERR39 DC C'PAGE WIDTH OR HEIGHT=0 OR > 132.ITEM IGNORED'	02720000
ERR40 DC C'SP OR I VALUE=0 OR >9.ITEM IGNORED'	02730000
ERR41 DC C'PAGEN OR MAXP VALUE=0 OR >9999.ITEM IGNORED'	02740000
ERR42 DC C'MAXL VALUE=0 OR >99.ITEM IGNORED'	02750000
ERR43 DC C'SUMMARY SPECIFIED ON CONTROL BREAK FIELD.PRIOR LIST ST.'	02760000
ERR44 DC C'KEY SUB-ST AND NO SAVE ST. WITH SFNAME PHRASE GIVEN'	02770000
ERR45 DC C'HEADING CONTAINS MORE THAN 14 CHARACTERS'	02780000
ERR46 DC C'INITIAL VALUE CONTAINS >16 CHARACTERS'	02790000

**OQLPARM (cont.)**

ERR47	DC	C'DEFINE EDIT STRING CONTAINS >3 CHARACTERS'	02800000
ERR48	DC	C'TWO FILE DESIGNATORS GIVEN AND CFNAME NOT ENTERED'	02810000
ERR49	DC	C'WARNING. FILE DOES NOT EXIST. FN,FT,AND MODE ARE:'	02820000
ERR50	DC	C'EDIT NOT SUPPORTED IN THIS VERSION'	02830000
ERR51	DC	C'NO STATEMENTS ENTERED AND ONLY TRAN OR OLD GIVEN'	02840000
ERR52	DC	C'END COMMAND NOT ENTERED'	02850000
ERR53	DC	C'SYNONYM TABLE FULL'	02860000
ERR54	DC	C'LINE CONTAINS MORE THAN 64 CHARACTERS'	02870000
ERR55	DC	C'ONLY START,QUIT,EDIT,RETAIN,OR CLEAR ALLOWED AFTER END'	02880000
ERR56	DC	C'MORE THAN 1 WHERE SUB-ST PER LIST OR SAVE ST'	02890000
ERR57	DC	C'NOT ENOUGH CORE FOR MARK IV. MINIMUM OF 384K REQUIRED'	02900000
ERR58	DC	C'LET EXPRESSION TOO LONG'	02910000
ERR59	DC	C'MORE THAN 9 DEC PLACES SPECIFIED'	02920000
ERR60	DC	C'FIELD LENGTH=0 OR > MAXIMUM ALLOWED LENGTH'	02930000
ERR61	DC	C'INVALID CONSOLE DESIGNATOR. CMD IGNORED'	02940000
ERR62	DC	C'TOO MANY CONTINUATION CARDS. JOB TERMINATED'	02950000
ERR63	DC	C'WARNING.NEXT LINE MUST NOT BE CONTINUED'	02960000
ERR64	DC	C'FILE DESIGNATOR TOO LONG'	02970000
ERR65	DC	C'PARTIAL FIELD START OR LENGTH=0 OR > 99'	02980000
ERR66	DC	C'SOURCE FILE DOES NOT EXIST. CMD IGNORED'	02990000
ERR67	DC	C'APOSTROPHE MISSING IN STRING'	03000000
ERR68	DC	C'WARNING.MORE THAN 1 REFERENCE TO SAME ITEM IN SUB-ST.'	03010000
ERR69	DC	C'CONTROL BREAK LEVEL=0 OR >9'	03020000
ERR70	DC	C'WARNING.BREAK(IF USED)MUST ASSIGN ALL CONTROL BRK LEVELS'	03030000
ERR71	DC	C'PARENTHESIS NESTED TO DEPTH > 9'	03040000
ERR72	DC	C'SAME NUMERATOR USED MORE THAN ONCE IN PERCENT/RATIO'	03050000
ERR73	DC	C'CHARACTER STRING SPLIT OVER 2 LINES'	03060000
ERR74	DC	C'CHARACTER STRING APPEARS IN ARITHMETIC EXPRESSION'	03070000
ERR75	DC	C'EDIT STRING GIVEN WITH C,V,OR E TYPE FIELDS'	03080000
ERR76	DC	C'DUPLICATE FILE ID IN SAVE STATEMENT'	03090000
ERR77	DC	C'MORE THAN 1 KEY SUB-ST. PER SAVE ST.'	03100000
ERR78	DC	C'TEMPORARY FIELD IS ALREADY DEFINED'	03110000
ERR79	DC	C'COPY OR EDIT COMMAND MUST BE LAST SENTENCE IN LINE'	03120000
ERR80	DC	C'INVALID DECIMAL NUMBER'	03130000
ERR81	DC	C'UNKNOWN KEYWORD. KEYWORD-VALUE PAIR IGNORED'	03140000
ERR82	DC	C'ILLEGAL VALUE FOR ITEM.KEYWORD-VALUE PAIR IGNORED'	03150000
ERR83	DC	C'BLOCKING FACTOR/BUFFER SIZE=0 OR > 9999'	03160000
ERR84	DC	C'ONLY 1 SAVE ST. PER SUBFILE MAY USE SFNAME'	03170000
ERR85	DC	C'MORE THAN 5 SUBFILE SELECTORS IN SAVEALL ST.'	03180000
ERR86	DC	C'NO PRECEDING LIST,SAVE OR SAVEALL ST.'	03190000
ERR87	DC	C'NO. OF SUBFILES SPECIFIED NE NO. OF FILE SELECTORS'	03200000
ERR88	DC	C'USE CMD NOT ENTERED FOR:'	03210000
ERR89	DC	C'TABLE LOOKUP FEATURE NOT SUPPORTED'	03220000
ERRN	EQU	*	03230000
	EJECT		03240000
*****			03250000
*			03260000
**** DO NOT MAKE CHANGES BEYOND HERE ****			03270000
*			03280000
*****			03290000
*	PRINT	OFF	03300000
CONTAB	DC	A(CONTABS,17,CONTABE)	03310000
*			03320000
	DS	0F	03330000
\$IECHO	DC	A(ECHO)	03340000
ITERM	DC	A(TERM)	03350000
CONTC	DC	AL1(CONTC)	03360000
*			03370000
QLMMSG	DC	A(MSGADR1-3, (MSGADR1-MSGADR1)/3)	03380000
MSGADR1	DC	AL3(ERR1)	03390000
	DC	AL3(ERR2)	03400000
	DC	AL3(ERR3)	03410000
	DC	AL3(ERR4)	03420000
	DC	AL3(ERR5)	03430000
	DC	AL3(ERR6)	03440000
	DC	AL3(ERR7)	03450000
	DC	AL3(ERR8)	03460000
	DC	AL3(ERR9)	03470000
	DC	AL3(ERR10)	03480000
	DC	AL3(ERR11)	03490000
	DC	AL3(ERR12)	03500000

### OQLPARM (cont.)

DC	AL3 (ERR13)	03510000
DC	AL3 (ERR14)	03520000
DC	AL3 (ERR15)	03530000
DC	AL3 (ERR16)	03540000
DC	AL3 (ERR17)	03550000
DC	AL3 (ERR18)	03560000
DC	AL3 (ERR19)	03570000
DC	AL3 (ERR20)	03580000
DC	AL3 (ERR21)	03590000
DC	AL3 (ERR22)	03600000
DC	AL3 (ERR23)	03610000
DC	AL3 (ERR24)	03620000
DC	AL3 (ERR25)	03630000
DC	AL3 (ERR26)	03640000
DC	AL3 (ERR27)	03650000
DC	AL3 (ERR28)	03660000
DC	AL3 (ERR29)	03670000
DC	AL3 (ERR30)	03680000
DC	AL3 (ERR31)	03690000
DC	AL3 (ERR32)	03700000
DC	AL3 (ERR33)	03710000
DC	AL3 (ERR34)	03720000
DC	AL3 (ERR35)	03730000
DC	AL3 (ERR36)	03740000
DC	AL3 (ERR37)	03750000
DC	AL3 (ERR38)	03760000
DC	AL3 (ERR39)	03770000
DC	AL3 (ERR40)	03780000
DC	AL3 (ERR41)	03790000
DC	AL3 (ERR42)	03800000
DC	AL3 (ERR43)	03810000
DC	AL3 (ERR44)	03820000
DC	AL3 (ERR45)	03830000
DC	AL3 (ERR46)	03840000
DC	AL3 (ERR47)	03850000
DC	AL3 (ERR48)	03860000
DC	AL3 (ERR49)	03870000
DC	AL3 (ERR50)	03880000
DC	AL3 (ERR51)	03890000
DC	AL3 (ERR52)	03900000
DC	AL3 (ERR53)	03910000
DC	AL3 (ERR54)	03920000
DC	AL3 (ERR55)	03930000
DC	AL3 (ERR56)	03940000
DC	AL3 (ERR57)	03950000
DC	AL3 (ERR58)	03960000
DC	AL3 (ERR59)	03970000
DC	AL3 (ERR60)	03980000
DC	AL3 (ERR61)	03990000
DC	AL3 (ERR62)	04000000
DC	AL3 (ERR63)	04010000
DC	AL3 (ERR64)	04020000
DC	AL3 (ERR65)	04030000
DC	AL3 (ERR66)	04040000
DC	AL3 (ERR67)	04050000
DC	AL3 (ERR68)	04060000
DC	AL3 (ERR69)	04070000
DC	AL3 (ERR70)	04080000
DC	AL3 (ERR71)	04090000
DC	AL3 (ERR72)	04100000
DC	AL3 (ERR73)	04110000
DC	AL3 (ERR74)	04120000
DC	AL3 (ERR75)	04130000
DC	AL3 (ERR76)	04140000
DC	AL3 (ERR77)	04150000
DC	AL3 (ERR78)	04160000
DC	AL3 (ERR79)	04170000
DC	AL3 (ERR80)	04180000
DC	AL3 (ERR81)	04190000
DC	AL3 (ERR82)	04200000
DC	AL3 (ERR83)	04210000

**OQLPARM (cont.)**

```

          DC      AL3 (ERR84)                                04220000
          DC      AL3 (ERR85)                                04230000
          DC      AL3 (ERR86)                                04240000
          DC      AL3 (ERR87)                                04250000
          DC      AL3 (ERR88)                                04260000
          DC      AL3 (ERR89)                                04270000
MSGADRN   DC      AL3 (ERRN)                                04280000
          EJECT                                              04290000
*****
* FILE DESIGNATOR AREA. HOLDS CURRENT VALUE OF ALL THE FILE 04300000
* DESIGNATORS WHICH CAN BE REFERENCED FROM QL. THE DEFAULT VALUES OF 04310000
* THESE DESIGNATORS IS GIVEN ABOVE.                                04320000
* THE LAYOUT OF EACH DESIGNATOR IS OPERATING SYSTEM DEPENDENT. 04330000
*                                                                    04340000
*****
*****
FDTAB     DC      A($$OLD,FDLN,LASTFD) USED TO COPY DEFAULTS FROM 04360000
          * AREA ABOVE. DONE BY REMOTE 4.                      04370000
          *
$FDLEN    DC      A (FDLN)                                    04380000
$$OLD     DC      (FDLN)C' '                                  04390000
$$NEW     DC      (FDLN)C' '                                  04400000
$$TRAN    DC      (FDLN)C' '                                  04410000
$$CORD1   DC      (FDLN)C' '                                  04420000
$$CORD2   DC      (FDLN)C' '                                  04430000
$$CORD3   DC      (FDLN)C' '                                  04440000
$$SUBF1   DC      (FDLN)C' '                                  04450000
$$SUBF2   DC      (FDLN)C' '                                  04460000
$$SUBF3   DC      (FDLN)C' '                                  04470000
$$SUBF4   DC      (FDLN)C' '                                  04480000
$$SUBF5   DC      (FDLN)C' '                                  04490000
$$SLIB    DC      (FDLN)C' '                                  04500000
$$LIB     DC      (FDLN)C' '                                  04510000
$$SUBF0   DC      (FDLN)C' '                                  04520000
$$LIST    DC      (FDLN)C' '                                  04530000
LASTFD    EQU     *-FDLN MUST FOLLOW LAST FILE DESIGNATOR      04540000
* END OF FILE DESIGNATOR AREA                                    04550000
          EJECT                                              04560000
*****
* MISCELLANEOUS FLAGS. THE INITIAL VALUES OF $TERM AND 04570000
* $ECHO ARE COPIED FROM ABOVE BY REMOTE 4.                      04580000
* FNAME,GLOSS,$PRORUN AND $RC ARE INITIALIZED DIRECTLY 04590000
* BY REMOTE 4.                                                  04600000
*****
*****
$$FNAME   DS      CL8 FILENAME (FOR RC CARD) .                  04610000
          * USED FOR GLOSSARY RUN                               04620000
          * 1ST BYTE=A,B OR 1.IS PUT ON FD                      04630000
          * CARD WHEN PRODUCING A GLOSSARY                      04640000
$PRORUN   DS      F =1 IF THIS IS A PROCESSING RUN              04650000
$RC       DS      F RETURN CODE                                 04660000
          *                                                      04670000
$ECHO     DS      F =1 IF SOURCE IS TO BE ECHO PRINTED         04680000
$TERM     DS      F FOR ONLINE SYSTEMS,GIVES THE MINIMUM      04690000
          * MESSAGE SEVERITY LEVEL FOR MESSAGES                04700000
          * PRINTED AT THE TERMINAL                             04710000
OPSYS     DC      AL3 (OPSYSD)                                  04720000
          *                                                      04730000
$MK4SPF   DC      A (TLU) TLU CAPABILITY FLAGS                 04740000
          ENTRY CONTAB,QLMMSG, $IECHO,OPSYS, ITERM,CONTCH      04750000
          ENTRY $$IOLD,$$INNEW,$$ITRAN,$$ICRD1,$$ICRD2,$$ICRD3 04760000
          ENTRY $$ISBF0,$$ISBF1,$$ISBF2,$$ILIST,$$ILIB,$$ISLIB 04770000
          ENTRY $$ISBF3,$$ISBF4,$$ISBF5                         04780000
          ENTRY $$QUERY,$$SRC,$$TEMP,$$TEMPE,MESSAGE,$$SVFD    04790000
          ENTRY $RC,$TERM,$PRORUN,$ECHO,$$FNAME,$$GLOSS        04800000
          ENTRY $$CORD1,$$CORD2,$$CORD3,$$SUBF3,$$SUBF4,$$SUBF5 04810000
          ENTRY $$OLD,$$NEW,$$TRAN,$$SUBF0,$$SUBF1,$$SUBF2      04820000
          ENTRY $$LIST,$$LIB,$$SLIB,FDTAB,$FDLEN,REPRC,M4CALL1,M4CALL2 04830000
          ENTRY GLOSSE,$MK4SPF                                  04840000
          END                                                    04850000
          *                                                      04860000
          *                                                      04870000
          *                                                      04880000
          *                                                      04890000
          *                                                      04900000

```



# Sample ISPF Startup CLIST

This appendix contains a sample CLIST that shows how you can make your VISION:Workbench for ISPF libraries available to the ISPF environment. You can then invoke this CLIST from the TSO prompt to start ISPF with the VISION:Workbench for ISPF option.

```
PROC 0
```

```
CONTROL LIST MSG
```

```
FREE F(SYSPROC ISPLLIB ISPMLIB ISPLLIB ISPSLIB ISPTLIB +
      ISPTABL ISPPROF M9LIST)
```

```
ALLOC F(SYSPROC) DA( 'SYS1.CMDPROC'          +
                    'ISR.ISPF.ISRCLIB'        +
                    'BUILDER.R140.SMPE.T.WBCLIST' ) SHR
```

```
ALLOC F(ISPLLIB) DA( 'BUILDER.R140.SMPE.T.BLSYSL' +
                    'INFORM40.LOADLIB'          +
                    'TRANSACT.TR075.GENLIB' ) SHR
```

```
ALLOC F(ISPMLIB) DA( 'BUILDER.R140.SMPE.T.WBMSG' +
                    'ISR.ISPF.ISRMLIB'          +
                    'ISP.ISPF.ISPMLIB' ) SHR
```

```
ALLOC F(ISPPLIB) DA( 'BUILDER.R140.SMPE.T.WBPANELS' +
                    'ISR.ISPF.ISRPLIB'          +
                    'ISP.ISPF.ISPPLIB' ) SHR
```

```
ALLOC F(ISPSLIB) DA( 'BUILDER.R140.SMPE.T.WBSKELS' +
                    'ISR.ISPF.ISRSLIB'          +
                    'ISP.ISPF.ISPSLIB' ) SHR
```

```
ALLOC F(ISPTLIB) DA( 'ISR.ISPF.ISRTLIB'          +
                    'ISP.ISPF.ISPTLIB' ) SHR
```

```
ALLOC F(ISPTABL) DA('ISR.ISPF.ISRTLIB')
```

```
ALLOC F(ISPPROF) DA('&SYSUID..ISPF.PROFILE')
```

```
/* NOTE: THE FOLLOWING ALLOCATE STATEMENT SHOWS HOW YOU CAN */
/* PRE-ALLOCATE YOUR WORKBENCH UTILITY LIST DATA SET.      */
/* THIS IS OPTIONAL. IF THIS DATA SET IS NOT PRE-ALLOCATED, */
/* WORKBENCH WILL DYNAMICALLY ALLOCATE IT WHEN NEEDED.      */
```

```
ALLOC F(M9LIST) DA('BUILDER.&SYSUID..M9LIST1') OLD
```

```
/* NOTE: THE FOLLOWING ALLOCATE STATEMENT SHOWS HOW WORKBENCH */
/* CUSTOMERS CAN ALLOCATE A FILE TAILORING OUTPUT DATA SET.  */
```

---

```
FREE FI(ISPFILE)

IF &SYSDSN('BUILDER.R140&SYSUID..FTOUTPUT') = OK THEN +
  ALLOC F(ISPFILE) DA('BUILDER.R140.&SYSUID..FTOUTPUT') SHR
ELSE +
  ALLOC F(ISPFILE) DA('BUILDER.R140.&SYSUID..FTOUTPUT') +
    NEW CATALOG UNIT(SYSDA) +
    SPACE(10,5) DIR(10) +
    DSORG(PO) RECFM(F B) LRECL(80) BLKSIZE(3120)

/* NOTE: THE FOLLOWING ALLOCATE STATEMENT SHOWS HOW WORKBENCH */
/* CUSTOMERS CAN ALLOCATE THE TABLE DATA SET USED BY THE IMPORT */
/* FACILITY. THE DATA SET MUST BE PREDEFINED AND CATALOGED. */
/* (THE "&SYSUID" ELEMENT IS REPLACED BY THE CURRENT USER ID.) */

  ALLOC F(DEFTLIB) DA('BUILDER.R140.&SYSUID..TLIB') SHR

/* NOW START ISPF USING THE ISPF PRIMARY MENU PANEL. */

ISPSTART PANEL(ISR@PRIM)

EXIT
```

# D Invocation Panels

The arrows on the following sample ISPF primary menu panel show how you can invoke VISION:Workbench for ISPF from your ISPF primary menu.

**XSR@PRIM**

```
-->> SAMPLE <<-- ISPF/PDF PRIMARY OPTION MENU VERSION n.n.n -----
%OPTION ==>_ZCMD
%
% 0 +ISPF PARMS - Specify terminal and user parameters +USERID - &ZUSER +
% 1 +BROWSE - Display source data or output listings +PROC - &ZLOGON
% 2 +EDIT - Create or change source data +PF KEYS - &ZKEYS
% 3 +UTILITIES - Perform utility functions +TERMINAL - &ZTERM
% 4 +FOREGROUND - Invoke language processors or script +TIME - &ZTIME
% 5 +BATCH - Submit job for language processing +JULIAN - &ZJDATE
% 6 +COMMAND - Enter TSO command or CLIST +DATE - &ZDATE
% 7 +DIALOG TEST - Perform dialog testing
% 8 +LM UTILITIES- Perform library administrator utility functions
% C +CHANGES - Display summary of changes for this release
% F +FILE-AID - FILE-AID data handling utilities
% M +LDSCS - Local Data Center Services Panels
% T +TUTORIAL - Display information about ISPF/PDF
--> % WB +Workbench -%VISION:Workbench Facility - Release 6.0
--> % BL +Builder -%VISION:Builder 14.0 Workbench
--> % TR +Transact -%Vision:Transact 7.5 Workbench
--> % IN +Inform -%VISION:Inform 4.0 Workbench (Definition Processor)
% X +EXIT - Terminate ISPF using log and list defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
.HELP = ISR00003
&ZPRIM = YES /* ALWAYS A PRIMARY OPTION MENU */
&ZHTOP = ISR00003 /* TUTORIAL TABLE OF CONTENTS */
&ZHINDEX = ISR91000 /* TUTORIAL INDEX - 1ST PAGE */
VPUT (ZHTOP,ZHINDEX) PROFILE
--> % &M9PRODUCT = 'Workbench'
)PROC
&ZSEL = TRANS( TRUNC(&ZCMD,'.')
0,'PANEL(ISPOPTA)'
1,'PGM(ISRBRO) PARM(ISRBRO01)'
2,'PGM(ISREDIT) PARM(P,ISREDM01)'
3,'PANEL(ISRUTIL)'
4,'PANEL(ISRFPA)'
5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
6,'PGM(ISRPTC)'
7,'PGM(ISRYXDR) NOCHECK'
8,'PANEL(ISRLPRIM)'
```

[illegible]

# Skeleton and User Panel Listings

This appendix contains the following skeleton and user panel listings:

- [M9BGUPNL – VISION:Builder Batch Job Submission User Panel on page E-1](#)
- [M9BGTS – VISION:Builder Batch Job Submission Skeleton on page E-2](#)
- [M9FGUPNL – VISION:Builder Foreground Job Execution User Panel on page E-5](#)
- [M9FGTS – VISION:Builder Foreground Job Execution Skeleton on page E-6](#)
- [M9GCTPU2 – VISION:Transact Batch Job Submission User Panel on page E-12](#)
- [M9GCTSBG – VISION:Transact Batch Job Submission Skeleton on page E-12](#)
- [M9GCTPU1 – VISION:Transact Foreground Job Execution User Panel on page E-16](#)
- [M9GCTSFG – VISION:Transact Foreground Job Execution Skeleton on page E-16](#)

## M9BGUPNL – VISION:Builder Batch Job Submission User Panel

```
)ATTR
+ TYPE(TEXT)      INTENS(LOW)  SKIP(&SKIPVAR)
% TYPE(TEXT)      INTENS(HIGH)  SKIP(&SKIPVAR)
  TYPE(INPUT)     INTENS(HIGH)  CAPS(ON)  JUST(LEFT)
^ TYPE(INPUT)     INTENS(NON)  CAPS(ON)  JUST(LEFT)
)BODY EXPAND(||)
%BATUPANL - &TMPIN |-|
%COMMAND ==>_ZCMD
+
+Enter%END+to process using the option selected on the BATCHOPT panel.
+Enter%CANCEL+command to terminate processing this member.
+
+Enter the name of the%VISION:Builder+ and %COMLIB+program LOAD LIBRARIES
+Builder LOADLIB %==> M9BGLLIB +
+COMLIB LOADLIB %==> M9BGLL2 +
+
+Enter the%VISION:Builder+region size (example 1024K)
+Builder REGION %==>_Z +
+
+Enter the name of the%SORT+program LOAD LIBRARY
+SORT LOADLIB %==>_M9BGSRT +
+
+SORT SPACE %==> Z + Number of SORTWORK CYLINDERS
+SORT UNIT %==>_M9BGSUNT+ SORTWORK UNIT type (example SYSDA)
+
+JOB statement information:
+ %==>_JOBREC1
```

---

## M9BGUPNL – VISION:Builder Batch Job Submission User Panel (cont.)

```
+ %===> _JOBREC2
+ %===> _JOBREC3
) INIT
  .ZVARS = ' (M9REGION M9BGSSP) '
  &ZCMD = &Z
  .HELP = M9BOTPU1
) PROC
VPUT (M9BGLLIB M9BGLL2 M9REGION M9BGSRT M9BGSSP M9BGSUNT) PROFILE
VPUT (JOBREC1 JOBREC2 JOBREC3) PROFILE
) END
```

## M9BGTS – VISION:Builder Batch Job Submission Skeleton

```
) CM
) CM THIS IS A SAMPLE ISPF FILE TAILORING SKELETON FOR USE WITH
) CM WORKBENCH RELEASE 6.0. IT WILL GENERATE MVS JCL FOR 1-STEP OR
) CM 3-STEP, SORT OR NOSORT BUILDER APPLICATION. THIS FILE TAILORING
) CM SKELETON IS INTENDED TO BE USED WITH THE SAMPLE USER PANEL
) CM 'M9BGUPNL' THAT HAS BEEN PROVIDED IN YOUR WORKBENCH PANEL LIBRARY.
) CM
) CM THERE ARE 4 TYPES OF VARIABLES (WORDS PRECEDED BY AMPERSANDS) USED
) CM IN THIS SKELETON INCLUDING:
) CM   - VARIABLES FROM THE USER PANEL M9BGUPNL
) CM   YOU CAN CHANGE THESE
) CM   - VARIABLES SET BY WORKBENCH
) CM   YOU CANNOT CHANGE THESE
) CM   - ISPF SYSTEM VARIABLES
) CM   - LOCAL VARIABLES THAT ARE SET AND USED DURING FILE TAILORING
) CM
) CM THE VARIABLES FROM THE USER PANEL INCLUDE:
) CM VARIABLE                USAGE
) CM
) CM &JOBREC1,&JOBREC2,&JOBREC3  MVS JOB STATEMENT INFORMATION
) CM &M9BGLLIB                 VISION:UILDER LOAD LIBRARY NAME
) CM &M9BGLL2                  COMLIB LOAD LIBRARY NAME
) CM &M9REGION                 REGION SIZE
) CM &M9BGSRT                  SORT PROGRAM LOAD LIBRARY
) CM &M9BGSUNT                 SORTWORK DATA SET UNIT TYPE
) CM &M9BGSSP                  SORTWORK DATA SET SPACE AMOUNT
) CM
) CM WORKBENCH RESERVED VARIABLE NAMES INCLUDE:
) CM VARIABLE                USAGE
) CM
) CM &M4DDNAM                   DATA DEFINITION NAME
) CM &M4DSN                    DATA SET NAME
) CM &M4DISP                   DATA SET STATUS AND DISPOSITION
) CM &M4VOL                     VOLUME SERIAL NUMBER
) CM &M4UNIT                    UNIT TYPE
) CM &M4DDOVERRIDE              OVERRIDE DATA DEFINITION NAME
) CM &M4RUNTYP                 VISION:UILDER RUN TYPE
) CM &PRJ1                      PDF LIBRARY PROJECT NAME
) CM &LIB1 THRU LIB4            PDF LIBRARY GROUP NAMES
) CM &TYP1                      PDF LIBRARY TYPE NAME
) CM &DSN                       'OTHER' PARTITIONED OR SEQUENTIAL FILE
) CM &MEMNAM                    MEMBER NAME
) CM &TMPIN                     QUALIFIED DSN FOR 'OTHER' DSN
) CM
) CM ISPF SYSTEM VARIABLES NAMES INCLUDE:
```

---

## M9BGTS – VISION:Builder Batch Job Submission Skeleton (cont.)

```
)CM VARIABLE                USAGE
)CM
)CM  &ZLLIB                  PDF LIBRARY GROUP NUMBER (1-4)
)CM  &Z                      A VARIABLE WHOSE VALUE IS NULL
)CM
)CM
&JOBREC1
&JOBREC2
&JOBREC3
//JOBLIB DD DSN=&M9BGLLIB,DISP=SHR
//      DD DSN=&M9BGLL2,DISP=SHR
//*
)SEL  &M9REGION ^= &Z
)SET  M9REG = ,REGION=&M9REGION
)ENDSEL
//MK4 EXEC PGM=MARKIV&M9REG
)CM
)CM LOOP THROUGH THE ISPF TABLE OF DATA SET CHARACTERISTICS ENTERED
)CM ON THE 'BATCHGEN' PANEL GENERATING APPROPRIATE DD STATEMENTS
)CM FOR EACH FILE TO BE USED IN THIS JOB.
)CM
)CM ***** BEGIN DD STATEMENT LOOP *****
)DOT DDNAMTB
)SEL  &M4DDOVER ^= &Z
)SET  M4DDNAM = &M4DDOVER
)ENDSEL
)CM
)CM SAVE M4LIST DSN AND UNIT SO PROPER DD STATEMENTS CAN BE GENERATED
)CM IN LATER STEPS OF A 3STEP RUN.
)CM
)SEL  &M4DDNAM = M4LIST
)SET  M4LSTDSN = &M4DSN
)SET  M4LSTUNT = &M4UNIT
)ENDSEL
)CM
)CM GENERATE DD STATEMENTS FOR SYSOUT DATA SETS
)CM
)SEL  &M4UNIT = SYSOUT
//&M4DDNAM DD SYSOUT=(&M4DSN)
)ENDSEL
)CM
)CM GENERATE DD STATEMENTS FOR NON-SYSOUT DATA SETS
)CM
)SET  M4DSN1 = &Z
)SEL  &M4DSN = &Z
)SET  M4DSN1 = DSN=NULLFILE
)ENDSEL
)SEL  &M4DSN ^= &Z
)SET  M4DSN1 = DSN=&M4DSN
)ENDSEL
)SEL  &M4UNIT ^= SYSOUT
)SEL  &M4DISP = NEW | &M4DISP = NEW,CATLG | &M4DISP = NEW,PASS
)SET  M4UNIT1 = &Z
)SET  M4VOL1 = &Z
)SEL  &M4UNIT ^= &Z
)SET  M4UNIT1 = ,UNIT=&M4UNIT
```

---

## M9BGTS – VISION:Builder Batch Job Submission Skeleton (cont.)

```
)ENDSEL
)SEL  &M4VOL ^= &Z
)SET  M4VOL1 = ,VOL=SER=&M4VOL
)ENDSEL
//&M4DDNAM DD &M4DSN1,DISP=(&M4DISP),
//          SPACE=(TRK,(5,5))&M4UNIT1&M4VOL1
)ENDSEL
)SEL &M4DISP ^= NEW && &M4DISP ^= NEW,CATLG && &M4DISP ^= NEW,PASS
)SET M4DISP1 = &Z
)SEL &M4DISP ^= &Z
)SET  M4DISP1 = ,DISP=(&M4DISP)
)ENDSEL
//&M4DDNAM DD &M4DSN1&M4DISP1
)ENDSEL
)ENDSEL
)ENDDOT
)CM  ***** END DD STATEMENT LOOP *****
)CM
)CM  ALLOCATE M4INPUT:
)CM  FOR PHYSICAL SEQUENTIAL DATA SETS JUST ALLOCATE THE DSN
)CM  FOR PDF LIBRARIES (VARIABLE ZLLIB = 1-4) BUILD A DSN STRING
)CM  WHICH NAMES THE APPROPRIATE PDF LIB AND MEMBER
)CM  FOR 'OTHER' PDS BUILD A DSN STRING WHICH NAMES THE LIB AND MEMBER
)CM
)SET M4INDSN = &Z
)SEL  &DSN = &Z
)SEL  &ZLLIB = 1
)SET  M4INDSN = &PRJ1..&LIB1..&TYP1(&MEMNAM)
)ENDSEL
)SEL  &ZLLIB = 2
)SET  M4INDSN = &PRJ1..&LIB2..&TYP1(&MEMNAM)
)ENDSEL
)SEL  &ZLLIB = 3
)SET  M4INDSN = &PRJ1..&LIB3..&TYP1(&MEMNAM)
)ENDSEL
)SEL  &ZLLIB = 4
)SET  M4INDSN = &PRJ1..&LIB4..&TYP1(&MEMNAM)
)ENDSEL
//M4INPUT DD DSN=&M4INDSN,DISP=SHR
)ENDSEL
)SEL  &DSN ^= &Z
//M4INPUT DD DSN=&TMPIN,DISP=SHR
)ENDSEL
)CM
)CM  ADD SORT DD STATEMENTS IF THIS IS A 1-STEP RUN
)CM
)SEL &M4RUNTYP = 1STEP
//SORTLIB DD DSN=&M9BGSRT,DISP=SHR
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=&M9BGSUNT,SPACE=(CYL,&M9BGSSP,,CONTIG)
//SORTWK02 DD UNIT=&M9BGSUNT,SPACE=(CYL,&M9BGSSP,,CONTIG)
//SORTWK03 DD UNIT=&M9BGSUNT,SPACE=(CYL,&M9BGSSP,,CONTIG)
)ENDSEL
//*
)CM
)CM  ADD JCL FOR SORT AND REPORT STEPS IF THIS IS A 3-STEP RUN
```

---

## M9BGTS – VISION:Builder Batch Job Submission Skeleton (cont.)

```
)CM
)SEL &M4RUNTYP = 3STEP
//SORT      EXEC  PGM=SORT
//SORTLIB    DD  DSN=&M9BGSRT,DISP=SHR
//SYSOUT     DD  SYSOUT=*
//SYSIN      DD  DSN=*.MK4.M4SORT,DISP=SHR
//SORTIN     DD  DSN=*.MK4.M4REPO,DISP=(OLD,PASS)
//SORTOUT    DD  DSN=&&&REPI,UNIT=SYSDA,SPACE=(TRK,(10,10)),
//            DISP=(NEW,PASS)
//SORTWK01   DD  UNIT=&M9BGSUNT,SPACE=(CYL,&M9BGSSP,,CONTIG)
//SORTWK02   DD  UNIT=&M9BGSUNT,SPACE=(CYL,&M9BGSSP,,CONTIG)
//SORTWK03   DD  UNIT=&M9BGSUNT,SPACE=(CYL,&M9BGSSP,,CONTIG)
/*
)CM
)CM      ADD JCL FOR REPORT STEP
)CM
//REPT      EXEC  PGM=MARKIV&M9REG
)SEL &M4LSTUNT = SYSOUT
//M4LIST    DD  SYSOUT=&M4LSTDSN
)ENDSEL
)SEL &M4LSTUNT ^= SYSOUT
//M4LIST    DD  DSN=&M4LSTDSN,DISP=MOD
)ENDSEL
//M4REPI    DD  DSN=*.SORT.SORTOUT,DISP=(OLD,PASS)
//M4INPUT   DD  *
REPTRUN RC          S
/*
)ENDSEL
//
```

## M9FGUPNL – VISION:Builder Foreground Job Execution User Panel

```
)ATTR
+ TYPE(TEXT)    INTENS(LOW)  SKIP(&SKIPVAR)
% TYPE(TEXT)    INTENS(HIGH) SKIP(&SKIPVAR)
_ TYPE(INPUT)   INTENS(HIGH) CAPS(ON) JUST(LEFT)
^ TYPE(INPUT)   INTENS(NON)  CAPS(ON) JUST(LEFT)
)BODY EXPAND(||)
%FORUPANL- &TMPIN |-|
%COMMAND ==>_ZCMD
+
+Enter%END+to process using the option selected on the FOREOPTS panel.
+Enter%CANCEL+command to terminate processing this member.
+
+Enter the name of the VISION:Builder LOAD LIBRARY
+VISION:Builder LOADLIB %==>_M9FGLLIB                                     +
+
+Enter the name of the SORT program LOAD LIBRARY
+SORT  LOADLIB %==>_M9FGSRT                                              +
+
+
+SORT SPACE      %==>_Z  +          Number of SORTWORK CYLINDERS
+SORT UNIT       %==>_M9FGSUNT+      SORTWORK UNIT type ( example%SYSDA+)
```

---

## M9FGUPNL – VISION:Builder Foreground Job Execution User Panel (cont.)

```
+
)INIT
    &ZCMD = &Z
    .ZVARS = ' (M9FGSSP)'
    .HELP = M9FOTPU1
)PROC
VPUT (M9FGLLIB M9FGSRT M9FGSSP M9FGSUNT) PROFILE
)END
```

## M9FGTS – VISION:Builder Foreground Job Execution Skeleton

```
)CM
)CM THIS IS A SAMPLE ISPF FILE TAILORING SKELETON FOR USE WITH
)CM WORKBENCH RELEASE 6.0. IT WILL GENERATE CLISTS FOR 1 STEP OR
)CM 3 STEP, SORT OR NOSORT, BUILDER APPLICATION. THIS FILE TAILORING
)CM SKELETON IS INTENDED TO BE USED WITH THE SAMPLE USER PANEL
)CM 'M9FGUPNL' WHICH HAS BEEN PROVIDED IN YOUR WORKBENCH PANEL LIB.
)CM
)CM THERE ARE 4 TYPES OF VARIABLES (WORDS PRECEDED BY AMPERSANDS) USED
)CM IN THIS SKELETON INCLUDING:
)CM     - VARIABLES FROM THE USER PANEL
)CM     YOU CAN CHANGE THESE
)CM     - VARIABLES SET BY WORKBENCH
)CM     YOU CANNOT CHANGE THESE
)CM     - ISPF SYSTEM VARIABLES
)CM     - LOCAL VARIABLES THAT ARE SET AND USED DURING FILE TAILORING
)CM
)CM THE VARIABLES FROM THE USER PANEL INCLUDE:
)CM VARIABLE                               USAGE
)CM
)CM &M9FGLLIB                               VISION:BUILDER LOAD LIBRARY NAME
)CM &M9FGSRT                                SORT PROGRAM LOAD LIBRARY
)CM &M9FGSUNT                                SORTWORK DATA SET UNIT TYPE
)CM &M9FGSSP                                SORTWORK DATA SET SPACE AMOUNT
)CM
)CM WORKBENCH RESERVED VARIABLE NAMES INCLUDE:
)CM VARIABLE                               USAGE
)CM
)CM &M4DDNAM                                DATA DEFINITION NAME
)CM &M4DSN                                  DATA SET NAME
)CM &M4DISP                                DATA SET STATUS AND DISPOSITION
)CM &M4VOL                                  VOLUME SERIAL NUMBER
)CM &M4UNIT                                  UNIT TYPE
)CM &M4DDOVER                               OVERRIDE DATA DEFINITION NAME
)CM &M4RUNTYP                               VISION:BUILDER RUN TYPE
)CM &PRJ1                                   PDF LIBRARY PROJECT NAME
)CM &LIB1 THRU LIB4                         PDF LIBRARY GROUP NAMES
)CM &TYP1                                   PDF LIBRARY TYPE NAME
)CM &DSN                                    'OTHER' PARTITIONED OR SEQUENTIAL FILE
)CM &MEMNAM                                 MEMBER NAME
)CM
)CM ISPF SYSTEM VARIABLE NAMES INCLUDE:
)CM VARIABLE                               USAGE
)CM
)CM &ZLLIB                                  PDF LIBRARY GROUP NUMBER (1-4)
```

---

## M9FGTS – VISION:Builder Foreground Job Execution Skeleton (cont.)

```
)CM  &Z                                A VARIABLE WHOSE VALUE IS NULL
)CM
PROC 0
CONTROL LIST MSG
/*                                     */
/* ALLOCATE VISION:UILDER FILES      */
/*                                     */
)CM
)CM  LOOP THRU THE TABLE OF FILES TO BE USED IN THIS APPLICATION
)CM  AND ALLOCATE EACH FILE AS SPECIFIED
)CM
SET &M4RC = 0
)DOT DDNAMTB
)CM
)CM  SAVE M4REPO DSN SO IT CAN BE ALLOCATED AS SORTIN IN SORT STEP
)CM  OF A 3-STEP RUN.
)CM
)SEL &M4DDNAM = M4REPO
)SET M4REPDSN = &M4DSN
)ENDSEL
)CM
)CM  SAVE M4LIST DSN SO IT CAN BE ALLOCATED AS SYSOUT IN SORT AND REP
)CM  STEPS OF A 3-STEP RUN.
)CM
)SEL &M4DDNAM = M4LIST
)SET M4LSTDSN = &M4DSN
)SET M4LSTUNT = &M4UNIT
)ENDSEL
)CM
)CM  SAVE M4SORT DSN SO IT CAN BE ALLOCATED AS SYSIN IN SORT STEP
)CM  OF A 3-STEP RUN.
)CM
)SEL &M4DDNAM = M4SORT
)SET M4SRITDSN = &M4DSN
)ENDSEL
)CM
)CM  CHANGE THE DDNAME TO THE OVERRIDE DDNAME WHERE APPLICABLE
)CM
)SEL &M4DDDOVER ^= &Z
)SET  M4DDNAM = &M4DDDOVER
)ENDSEL
FREE FI(&M4DDNAM)
)CM
)CM  ALLOCATE SYSOUT DATA SETS
)CM
)SEL &M4UNIT = SYSOUT
)SEL  &M4DSN ^= *
ALLOC FI(&M4DDNAM) SYSOUT(&M4DSN)
)ENDSEL
)SEL  &M4DSN = *
ALLOC FI(&M4DDNAM) DA(&M4DSN)
)ENDSEL
)ENDSEL
)CM
)CM  ALLOCATE NON-SYSOUT DATA SETS.
)CM  CREATE UNIT, DSN, AND VOLUME STRINGS.
```

---

## M9FGTS – VISION:Builder Foreground Job Execution Skeleton (cont.)

```
)CM
)SEL &M4UNIT ^= SYSOUT
)SET M4UNIT1 = &Z
)SET M4DSN1 = &Z
)SET M4VOL1 = &Z
)CM
)SEL &M4UNIT ^= &Z
)SET M4UNIT1 = UNIT(&M4UNIT)
)ENDSEL
)CM
)SEL &M4VOL ^= &Z
)SET M4VOL1 = VOLUME(&M4VOL)
)ENDSEL
)CM
)SEL &M4DSN ^= &Z
)SET M4DSN1 = DA(&M4DSN)
)ENDSEL
)CM
)SEL &M4DSN = &Z
)SET M4DSN = DUMMY
)SET M4UNIT1 = &Z
)SET M4DSN1 = &Z
)SET M4VOL1 = &Z
)SET M4DISP = &Z
)ENDSEL
)CM
)CM ALLOCATE DUMMY DATA SETS.
)CM
)SEL &M4DSN = DUMMY
ALLOC FI(&M4DDNAM) &M4DSN
)ENDSEL
)CM
)CM ALLOCATE NEW NON-SYSOUT DATA SETS.
)CM
)SEL &M4DSN ^= DUMMY
)SEL &M4DISP = NEW | &M4DISP = NEW,CATALOG
ALLOC FI(&M4DDNAM) &M4DSN1 &M4DISP &M4UNIT1 &M4VOL1 +
SPACE(5 5) TRACK
)ENDSEL
)CM
)SEL &M4DISP = NEW,DELETE
ALLOC FI(&M4DDNAM) &M4DSN1 &M4DISP &M4UNIT1 &M4VOL1 +
SPACE(5 5) TRACK
)ENDSEL
)CM
)CM ALLOCATE NON-NEW NON-SYSOUT DATA SETS.
)CM
)SEL &M4DISP ^= NEW && &M4DISP ^= NEW,CATALOG
)SEL &M4DISP ^= NEW,DELETE
ALLOC FI(&M4DDNAM) &M4DSN1 &M4DISP &M4UNIT1 &M4VOL1
)ENDSEL
)ENDSEL
)ENDSEL
)ENDSEL
)ENDDOT
)CM
```

---

## M9FGTS – VISION:Builder Foreground Job Execution Skeleton (cont.)

```
)CM ALLOCATE M4INPUT:
)CM FOR PHYSICAL SEQUENTIAL DATA SETS JUST ALLOCATE THE DSN
)CM FOR PDF LIBRARIES (VARIABLE ZLLIB = 1-4) BUILD A DSN STRING
)CM WHICH NAMES THE APPROPRIATE PDF LIB AND MEMBER
)CM FOR 'OTHER' PDS BUILD A DSN STRING WHICH NAMES THE LIB AND MEMBER
)CM FOR PDS AND PDF LIBS QUALIFICATION, QUOTES MUST BE CONSIDERED
)CM
FREE FI(M4INPUT)
)SET M4INDSN = &Z
)SEL &ORGVAR = PS
ALLOC FI(M4INPUT) DA(&DSN) SHR
)ENDSEL
)SEL &ORGVAR = PO
)SEL &DSN = &Z
)SEL &ZLLIB = 1
)SET M4INDSN = ' &PRJ1..&LIB1..&TYP1(&MEMNAM) '
)ENDSEL
)SEL &ZLLIB = 2
)SET M4INDSN = ' &PRJ1..&LIB2..&TYP1(&MEMNAM) '
)ENDSEL
)SEL &ZLLIB = 3
)SET M4INDSN = ' &PRJ1..&LIB3..&TYP1(&MEMNAM) '
)ENDSEL
)SEL &ZLLIB = 4
)SET M4INDSN = ' &PRJ1..&LIB4..&TYP1(&MEMNAM) '
)ENDSEL
ALLOC FI(M4INPUT) DA(&M4INDSN) SHR
)ENDSEL
)SEL &DSN ^= &Z
SET &&DSN = &&STR(&DSN2)
IF &&SUBSTR(1:1,&&DSN) = &&STR(') THEN DO
    SET &&HLDDSN = &&SUBSTR(2:&&LENGTH(&&DSN)-1,&&DSN)
    SET &&HLDDSN = &&STR(' &&HLDDSN.(&MEMNAM) ' )
END
ELSE DO
    SET &&HLDDSN = &&STR(&&DSN.(&MEMNAM) )
END
ALLOC FI(M4INPUT) DA(&&HLDDSN) SHR
)ENDSEL
)ENDSEL
)CM
)CM CHECK TO SEE IF THIS IS A 1-STEP OR 3-STEP RUN AND IF SO
)CM ADD JCL FOR SORT
)CM
/*
/* ALLOCATE SORT FILES
/*
)SEL &M4RUNTYP = 1STEP | &M4RUNTYP = 3STEP
FREE FI(SORTLIB SYSOUT SORTWK01 SORTWK02 SORTWK03)
ALLOC FI(SORTLIB) DA(&M9FGSRT) SHR
)CM
)CM ALLOCATE SYSOUT FOR SORT. IF M4LIST WAS A SYSOUT DATA SET, THEN
)CM ALLOCATE SYSOUT THE SAME. IF M4LIST WAS NOT A SYSOUT DATA SET,
)CM THEN ALLOCATE SYSOUT TO A NEW UNNAMED (AND THUS TEMPORARY) FILE
)CM TO AVOID I/O ERRORS FROM DCB CONFLICTS.
)CM
```

---

## M9FGTS – VISION:Builder Foreground Job Execution Skeleton (cont.)

```
)SEL &M4LSTUNT = SYSOUT
)SEL &M4LSTDSN ^= *
ALLOC FI(SYSOUT) SYSOUT(&M4LSTDSN)
)ENDSEL
)SEL &M4LSTDSN = *
ALLOC FI(SYSOUT) DA(&M4LSTDSN)
)ENDSEL
)ENDSEL
)SEL &M4LSTUNT ^= SYSOUT
ALLOC FI(SYSOUT) NEW UNIT(SYSDA)
)ENDSEL
ALLOC FI(SORTWK01) UNIT(&M9FGSUNT) SPACE(&M9FGSSP) CYLINDERS
ALLOC FI(SORTWK02) UNIT(&M9FGSUNT) SPACE(&M9FGSSP) CYLINDERS
ALLOC FI(SORTWK03) UNIT(&M9FGSUNT) SPACE(&M9FGSSP) CYLINDERS
)ENDSEL
/*                                                    */
/* EXECUTE THE VISION:UILDER PROCESS STEP                */
/*                                                    */
SET &M9FGLLIB = &&STR(&M9FGLLIB)
IF &&SUBSTR(1:1,&&M9FGLLIB) = &&STR(') THEN DO
    SET &M9FGMPGM = &&SUBSTR(2:&&LENGTH(&&M9FGLLIB)-1,&&M9FGLLIB)
    SET &M9FGMPGM = &&STR(' &M9FGMPGM.(MARKIV)')
END
ELSE DO
    SET &M9FGMPGM = &&STR(&&M9FGLLIB.(MARKIV))
END
CALL &M9FGMPGM
/*                                                    */
/* IF BUILDER PROCESS STEP FAILED SKIP THE SORT AND REPORT STEPS */
/*                                                    */
IF &&LASTCC ^= 0 THEN DO
    SET &M4RC = 8
    GOTO EXIT
END
)SEL &M4RUNTYP = 3STEP
/*                                                    */
/* EXECUTE THE SORT STEP FOR 3 STEP RUNS                */
/*                                                    */
FREE FI(SORTIN SORTOUT SYSIN)
)CM
)CM ALLOCATE SORTIN TO THE M4REPO DSN CREATED IN PROCESS STEP
)CM ALLOCATE SORTOUT TO A NEW DATA SET WHICH WILL BE DELETED LATER
)CM ALLOCATE SYSIN (SORT CNTL STMTS) TO THE M4SORT DSN CREATED IN
)CM THE PROCESSING STEP
)CM
ALLOC FI(SORTIN) DA(&M4REPDSN) SHR
ALLOC FI(SORTOUT) DA(&ZUSER..REPI) NEW +
    UNIT(SYSDA) SPACE(5 5) TRACK
ALLOC FI(SYSIN) DA(&M4SRTDSN) SHR
SET &M9FGSRT = &&STR(&M9FGSRT)
IF &&SUBSTR(1:1,&&M9FGSRT) = &&STR(') THEN DO
    SET &M9FGSPGM = &&SUBSTR(2:&&LENGTH(&&M9FGSRT)-1,&&M9FGSRT)
    SET &M9FGSPGM = &&STR(' &M9FGSPGM.(SORT)')
END
ELSE DO
    SET &M9FGSPGM = &&STR(&&M9FGSRT.(SORT))
```

---

## M9FGTS – VISION:Builder Foreground Job Execution Skeleton (cont.)

```
END
CALL &M9FGSPGM
/*
/* IF THE SORT STEP FAILED SKIP THE REPORT STEP
/*
/*
IF &&LASTCC ^= 0 THEN DO
    SET &&M4RC = 8
    GOTO EXIT
END
/*
/* EXECUTE THE REPORT STEP
/*
/*
)CM
)CM ALLOCATE M4REPI TO THE SORTOUT DSN CREATED IN SORT STEP
)CM ALLOCATE M4INPUT TO A NEW TEMPORARY DATA SET
)CM OPEN THE M4INPUT DSN, WRITE THE REPORT RUN RC STATEMENT AND THEN
)CM CLOSE M4INPUT
)CM
FREE FI(M4REPI)
ALLOC FI(M4REPI) DA(&ZUSER..REPI) SHR
FREE FI(M4INPUT)
ALLOC FI(M4INPUT) NEW UNIT(SYSDA) SPACE(1) TRACK
OPENFILE M4INPUT OUTPUT
SET &&M4INPUT = REPTRUNCRC          S      A
PUTFILE M4INPUT
CLOSFILE M4INPUT
CALL &M9FGMPGM
IF &&LASTCC ^= 0 THEN DO
    SET &&M4RC = &&LASTCC
END
/*
/* FREE FILES
/*
/*
DEL &ZUSER..REPI NONVSAM
FREE FI(SORTIN SORTOUT SYSIN M4REPI)
)ENDSEL
)CM
)CM FREE ALLOCATIONS
)CM
EXIT:FREE FI(M4INPUT)
)SEL &M4RUNTYP = 1STEP | &M4RUNTYP = 3STEP
FREE FI(SORTLIB SYSOUT M4SORT)
FREE FI(SORTWK01 SORTWK02 SORTWK03)
)ENDSEL
)DOT DDNAMTB
)SEL &M4DDOVER ^= &Z
)SET M4DDNAM = M4DDOVER
)ENDSEL
FREE FI(&M4DDNAM)
)ENDDOT
EXIT CODE(&M4RC)
)CM END OF SKELETON
```

---

## M9GCTPU2 – VISION:Transact Batch Job Submission User Panel

```
)ATTR
+ TYPE(TEXT) INTENS(LOW) SKIP(&SKIPVAR)
% TYPE(TEXT) INTENS(HIGH) SKIP(&SKIPVAR)
_ TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT)
@ TYPE(INPUT) INTENS(LOW) CAPS(ON) JUST(LEFT) PADC(&PADVAR)
} TYPE(OUTPUT) INTENS(HIGH) SKIP(ON)
{ TYPE(OUTPUT) INTENS(LOW) SKIP(ON)
)BODY EXPAND(||)
%USRPNEL --- &GVITEMLM -|-|
%COMMAND ==> _ZCMD
+
+ENTER%END+TO PROCESS USING THE OPTION PREVIOUSLY ENTERED.
+ENTER%CANCEL+TO TERMINATE PROCESSING THIS MEMBER.
+
+
%ENTER THE NAME OF THE VISION:TRANSACT GEN LIBRARY:
+VISION:TRANSACT LOADLIB ==> _GCUSVL2 +
+
%ENTER THE NAME OF THE COMLIB LOAD LIBRARY:
+COMLIB LOADLIB ==> _GCUSLL2 +
+
%LINK OBJECT FILE? ==> Z + (YES OR NO)
+ LINK TO ==> _GCUSLKL2 +
+ INCLUDES FROM:
+ VISION:TRANSACT ==> _GCUSMKI2 +
+ MONITOR ==> _GCUSMNI2 +
)INIT
.HELP = M9GCTPH0
.ZVARS = ' (GCUSLNK) '
)PROC
VPUT (GCUSVL2 GCUSLL2 GCUSLNK GCUSLKL2 GCUSMKI2 GCUSMNI2) PROFILE
)END
```

## M9GCTSBG – VISION:Transact Batch Job Submission Skeleton

```
)CM
)CM THIS IS A SAMPLE ISPF FILE TAILORING SKELETON FOR USE WITH
)CM WORKBENCH RELEASE 6.0. IT WILL GENERATE MVS JCL FOR A TRANSACT
)CM BATCH RUN (APPGEN, DEF RUN, OR SSR). THIS FILE TAILORING
)CM SKELETON IS INTENDED TO BE USED WITH THE SAMPLE USER PANEL
)CM 'M9GCTPU2'. THIS PANEL HAS BEEN PROVIDED IN YOUR WORKBENCH
)CM PANEL LIBRARY.
)CM
)CM THERE ARE 4 TYPES OF VARIABLES (WORDS PRECEDED BY AMPERSANDS) USED
)CM IN THIS SKELETON INCLUDING:
)CM - VARIABLES FROM THE USER PANEL
)CM YOU CAN CHANGE THESE
)CM - VARIABLES SET BY WORKBENCH
)CM YOU CANNOT CHANGE THESE
)CM - ISPF SYSTEM VARIABLES
)CM - LOCAL VARIABLES THAT ARE SET AND USED DURING FILE TAILORING
)CM
)CM THE VARIABLES FROM THE USER PANEL INCLUDE:
)CM VARIABLE USAGE
```

---

## M9GCTSBG - VISION:Transact Batch Job Submission Skeleton (cont.)

```
)CM
)CM  &GCUSVL2          VISION:TRANSACT GEN LIBRARY
)CM  &GCUSLL2          COMLIB LOAD LIBRARY
)CM  &GCUSLNK          IF THE OBJECT FILE SHOULD BE LINKED
)CM  &GCUSLKL2         LINK LIBRARY
)CM  &GCUSMKI2         VISION:TRANSACT INCLUDE LIBRARY
)CM  &GCUSMNI2         MONITOR INCLUDE LIBRARY
)CM
)CM  WORKBENCH RESERVED VARIABLE NAMES INCLUDE:
)CM  VARIABLE          USAGE
)CM
)CM  &GSDDNAM          DATA DEFINITION NAME
)CM  &GSDSN           DATA SET NAME
)CM  &GSDISP          DATA SET STATUS AND DISPOSITION
)CM  &GSVOLSER        VOLUME SERIAL NUMBER
)CM  &GSUNIT          UNIT TYPE
)CM  &GSALLOC         FILE ALLOCATION
)CM
)CM  ISPF SYSTEM VARIABLES NAMES INCLUDE:
)CM  VARIABLE          USAGE
)CM
)CM  &Z                A VARIABLE WHOSE VALUE IS NULL
)CM
)CM
)SEL  &GCBOJCL1 ^= &Z
&GCBOJCL1
)ENDSEL
)SEL  &GCBOJCL2 ^= &Z
&GCBOJCL2
)ENDSEL
)SEL  &GCBOJCL3 ^= &Z
&GCBOJCL3
)ENDSEL
)SEL  &GCBOJCL4 ^= &Z
&GCBOJCL4
)ENDSEL
)CM
)CM  INCLUDE EXECUTE AND STEPLIB STATEMENTS
)CM
//MISPF   EXEC PGM=MARKV,REGION=2M
//STEPLIB DD DSN=&GCUSVL2,DISP=SHR
//        DD DSN=&GCUSLL2,DISP=SHR
)CM
)CM  LOOP THROUGH THE ISPF TABLE OF DATA SET CHARACTERISTICS ENTERED
)CM  ON THE 'BATCHGEN' PANEL GENERATING APPROPRIATE DD STATEMENTS
)CM  FOR EACH FILE TO BE USED IN THIS JOB.
)CM
)CM  ***** BEGIN DD STATEMENT LOOP *****
)DOT M9GDFTB
)CM
)CM  CHECK IF DD STATEMENT IS FOR M5LIST
)CM
)SEL  &GSDDNAM = M5LIST
)CM
)CM  SET DEFAULT FOR SYSPRINT DD IN LINK STEP
)CM
```

---

## M9GCTSBG – VISION:Transact Batch Job Submission Skeleton (cont.)

```
)SET M5SYSOUT = A
)SEL &GSUNIT = SYSOUT
)SET M5SYSOUT = &GSDSN
)ENDSEL
)ENDSEL
)CM
)CM CHECK IF DD STATEMENT IS FOR M5PUNCH
)CM
)SEL &GSDDNAM = M5PUNCH
)SET M5PUNCH = &GSDSN
)ENDSEL
)CM
)CM CHECK IF DD STATEMENT IS FOR SYSOUT DATA SET
)CM
)SEL &GSUNIT = SYSOUT
//&GSDDNAM DD SYSOUT=&GSDSN
)ENDSEL
)CM
)CM GENERATE DD STATEMENTS FOR NON-SYSOUT DATA SETS
)CM
)CM DEFAULT UNIT TO NULL, IF SPECIFIED, SET IT
)CM
)SEL &GSUNIT ^= SYSOUT
)SET M5UNIT1 = &Z
)SEL &GSUNIT ^= &Z
)SET M5UNIT1 = UNIT=&GSUNIT
)ENDSEL
)CM
)CM DEFAULT VOLSER TO NULL, IF SPECIFIED, SET IT
)CM
)SET M5VSR1 = &Z
)SEL &GSVOLSER ^= &Z
)SET M5VSR1 = VOL=SER=&GSVOLSER
)ENDSEL
)CM
)CM DEFAULT ALLOCATION TO NULL, IF SPECIFIED, SET IT
)CM
)SET M5ALL1 = &Z
)SEL &GSALLOC ^= &Z
)SET M5ALL1 = SPACE=(&GSALLOC)
)ENDSEL
)CM
)CM CHECK COMMA PLACEMENTS
)CM
)SET M5C1 = &Z
)SET M5C2 = &Z
)SEL M5VSR ^= &Z | M5UNIT ^= &Z
)SET M5C1 = ,
)ENDSEL
)SEL M5UNIT ^= &Z
)SET M5C2 = ,
)ENDSEL
)CM
)CM USE THIS DD IF DD = DUMMY
)CM
)SEL &GSDSN = DUMMY
```

---

## M9GCTSBG - VISION:Transact Batch Job Submission Skeleton (cont.)

```
//&GSDDNAM DD DUMMY
)ENDSEL
)CM
)CM USE THIS DD IF ALLOCATION, VOLSER, OR UNIT IS SPECIFIED
)CM
)SEL &GSDSN ^= DUMMY
)SEL &M5ALL1 ^= &Z | &M5VSER1 ^= &Z | &M5UNIT ^= &Z
//&GSDDNAM DD DSN=&GSDSN,DISP=(&GSDISP),
)SEL &M5ALL1 ^= &Z
//          &M5ALL1&M5C1
)ENDSEL
)SEL &M5VSER1 ^= &Z
//          &M5VSER1&M5C2
)ENDSEL
)SEL &M5UNIT1 ^= &Z
//          &M5UNIT1
)ENDSEL
)ENDSEL
)ENDSEL
)CM
)CM USE THIS DD IF ALLOCATION, VOLSER, AND UNIT ARE NOT SPECIFIED
)CM
)SEL &GSDSN ^= DUMMY
)SEL &M5ALL1 = &Z && &M5VSER1 = &Z && &M5UNIT = &Z
//&GSDDNAM DD DSN=&GSDSN,DISP=(&GSDISP)
)ENDSEL
)ENDSEL
)CM
)CM END LOOP ON NON-SYSOUT DATA SET
)CM
)ENDSEL
)CM
)CM END LOOP ON TABLE
)CM
)ENDDOT
)CM
)CM SEE IF LINK IS REQUIRED
)CM
)SEL &GCDDRTYP = A | &GCDDRTYP = G
)SEL &GCUSLNK = Y | &GCUSLNK = YES
//*
//LINK EXEC PGM=HEWL,REGION=2M,
//          PARM='LET,LIST,XREF,RENT,REUS'
//SYSPRINT DD &M5SYSOUT
//SYSLMOD DD DSN=&GCUSLKL2,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIB DD DSN=&GCUSMKI2,DISP=SHR
//          DD DSN=&GCUSMNI2,DISP=SHR
//IMSLIB DD DSN=&GCUSMNI2,DISP=SHR
//SYSLIN DD DSN=&M5PUNCH,DISP=SHR
)ENDSEL
)ENDSEL
)CM
)CM ADD EOJ STATEMENT
)CM
//
```

---

## M9GCTPU1 – VISION:Transact Foreground Job Execution User Panel

```
)ATTR
+ TYPE(TEXT) INTENS(LOW) SKIP(&SKIPVAR)
% TYPE(TEXT) INTENS(HIGH) SKIP(&SKIPVAR)
_ TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT)
@ TYPE(INPUT) INTENS(LOW) CAPS(ON) JUST(LEFT) PADC(&PADVAR)
} TYPE(OUTPUT) INTENS(HIGH) SKIP(ON)
{ TYPE(OUTPUT) INTENS(LOW) SKIP(ON)
)BODY EXPAND(||)
%USRPNEL --- &GVITEMLM -|-|
%COMMAND ==> _ZCMD
+
+ENTER%END+TO PROCESS USING THE OPTION PREVIOUSLY ENTERED.
+ENTER%CANCEL+TO TERMINATE PROCESSING THIS MEMBER.
+
+
%ENTER THE NAME OF THE VISION:TRANSACT GEN LIBRARY:
+VISION:TRANSACT LOADLIB ==> _GCUSVL1 +
+
%LINK OBJECT FILE? ==> _Z + (YES OR NO)
+ LINK TO ==> _GCUSLKL1 +
+ INCLUDES FROM:
+ VISION:TRANSACT ==> _GCUSMKI1 +
+ MONITOR ==> _GCUSMNI1
)INIT
.HELP = M9GCTPH0
.ZVARS = ' (GCUSLNK) '
)PROC
VPUT (GCUSVL1 GCUSLNK GCUSLKL1 GCUSMKI1 GCUSMNI1) PROFILE
)END
```

## M9GCTSFG – VISION:Transact Foreground Job Execution Skeleton

```
)CM
)CM THIS IS A SAMPLE ISPF FILE TAILORING SKELETON FOR USE WITH
)CM WORKBENCH RELEASE 6.0. IT WILL GENERATE CLISTS FOR A TRANSACT
)CM FOREGROUND RUN (APPGEN, DEF RUN, OR SSR). THIS FILE TAILORING
)CM SKELETON IS INTENDED TO BE USED WITH THE SAMPLE USER PANEL
)CM 'M9GCTPU1'. THIS PANEL HAS BEEN PROVIDED IN YOUR WORKBENCH
)CM PANEL LIBRARY.
)CM
)CM THERE ARE 4 TYPES OF VARIABLES (WORDS PRECEDED BY AMPERSANDS) USED
)CM IN THIS SKELETON INCLUDING:
)CM - VARIABLES FROM THE USER PANEL
)CM YOU CAN CHANGE THESE
)CM - VARIABLES SET BY WORKBENCH
)CM YOU CANNOT CHANGE THESE
)CM - ISPF SYSTEM VARIABLES
)CM - LOCAL VARIABLES THAT ARE SET AND USED DURING FILE TAILORING
)CM
)CM THE VARIABLES FROM THE USER PANEL INCLUDE:
)CM VARIABLE USAGE
)CM
)CM &GCUSVL1 VISION:TRANSACT GEN LIBRARY
)CM &GCUSLNK IF THE OBJECT FILE SHOULD BE LINKED
```

---

## M9GCTSG - VISION:Transact Foreground Job Execution Skeleton (cont.)

```
)CM  &GCUSLKL1          LINK LIBRARY
)CM  &GCUSMKI1          VISION:TRANSACT INCLUDE LIBRARY
)CM  &GCUSMNI1          MONITOR INCLUDE LIBRARY
)CM
)CM  WORKBENCH RESERVED VARIABLE NAMES INCLUDE:
)CM  VARIABLE            USAGE
)CM
)CM  &GSDDNAM            DATA DEFINITION NAME
)CM  &GSDSN              DATA SET NAME
)CM  &GSDISP             DATA SET STATUS AND DISPOSITION
)CM  &GSVOLSER           VOLUME SERIAL NUMBER
)CM  &GSUNIT             UNIT TYPE
)CM  &GSALLOC            FILE ALLOCATION
)CM
)CM  ISPF SYSTEM VARIABLE NAMES INCLUDE:
)CM  VARIABLE            USAGE
)CM
)CM  &Z                  A VARIABLE WHOSE VALUE IS NULL
)CM
)CM
PROC 0
CONTROL LIST MSG
/*                                */
/* ALLOCATE FILES                */
/*                                */
)CM
)CM  LOOP THRU THE TABLE OF FILES TO BE USED IN THIS APPLICATION
)CM  AND ALLOCATE EACH FILE AS SPECIFIED
)CM
)DOT M9GDFTB
)CM
)CM  SAVE M5LIST DSN SO IT CAN BE ALLOCATED AS SYSLST IN THE LINK STEP
)CM
)SEL &GSDDNAM = M5LIST
)SET M5LIST = A
)SEL &GSUNIT = SYSOUT
)SET M5LIST = &GSDSN
)ENDSEL
)SEL &GSDSN = *
)SET M5LIST = *
)ENDSEL
)ENDSEL
)CM
)CM  SAVE M5PUNCH DSN SO IT CAN BE ALLOCATED AS SYSLIN IN THE LINK STEP
)CM
)SEL &GSDDNAM = M5PUNCH
)SET M5PUNCH = &GSDSN
)ENDSEL
FREE FI(&GSDDNAM)
)CM
)CM  ALLOCATE SYSOUT DATA SETS
)CM
)SEL &GSUNIT = SYSOUT
)SEL  &GSDSN ^= *
ALLOC FI(&GSDDNAM) SYSOUT(&GSDSN)
)ENDSEL
```

---

## M9GCTSFG – VISION:Transact Foreground Job Execution Skeleton (cont.)

```
)SEL    &GSDSN = *
ALLOC FI(&GSDDNAM) DA(&GSDSN)
)ENDSEL
)ENDSEL
)CM
)CM  ALLOCATE NON-SYSOUT DATA SETS.
)CM  CREATE UNIT, DSN, AND VOLUME STRINGS.
)CM
)SEL  &GSUNIT ^= SYSOUT
)SET  M5UNIT1 = &Z
)SET  M5DSN1  = &Z
)SET  M5VOL1  = &Z
)SEL  &M5UNIT ^= &Z
)SET  M5UNIT1 = UNIT(&GSUNIT)
)ENDSEL
)SEL  &GSDSN ^= &Z
)SET  M5DSN1 = DA(&GSDSN)
)ENDSEL
)SEL  &M5VOL ^= &Z
)SET  M5VOL1 = VOLUME(&M5VOL)
)ENDSEL
)CM
)CM  ALLOCATE NON-SYSOUT DATA SETS
)CM
ALLOC FI(&GSDDNAM) &GSALLOC &GSDISP &M5UNIT1 &M5VOL1 +
&M5DSN1
)ENDSEL
)ENDDOT
)CM
)CM  ALLOCATE STANDARD DATA SETS
)CM
/*                                          */
/* EXECUTE VISION:TRANSACTION          */
/*                                          */
SET &M5 = &&STR(&GCUSVL1)
IF &&SUBSTR(1:1,&M5) = &&STR(') THEN DO
    SET &M5 = &&SUBSTR(2:&&LENGTH(&M5)-1,&M5)
    SET &M5 = &&STR(' &M5.(MARKV)')
END
ELSE DO
    SET &M5 = &&STR(&M5.(MARKV))
END
CALL &M5
)CM
)CM  FREE DATA SETS
)CM
)DOT M9GDFTB
    FREE FI(&GSDDNAM)
)ENDDOT
)CM
)CM  CHECK FOR LINKEDIT STEP
)CM
)SEL  &GCDDRTYP = A && &GCUSLNK = YES
/*                                          */
/* LINKEDIT STEP                        */
/*                                          */
```

---

## M9GCTSG - VISION:Transact Foreground Job Execution Skeleton (cont.)

```
IF &&LASTCC = 0 THEN
FREE FI (SYSPRINT SYSLMOD SYSUT1 SYSLIB IMSLIB SYSLIN)
)SEL &M5LIST = *
ALLOC FI (SYSPRINT) DA (&M5LIST)
)ENDSEL
)SEL &M5LIST ^= *
ALLOC FI (SYSPRINT) SYSOUT (&M5LIST)
)ENDSEL
ALLOC FI (SYSLMOD) DA (&GCUSLKL1) SHR
ALLOC FI (SYSUT1) UNIT (SYSDA) SPACE (1 1) CYLINDERS
ALLOC FI (SYSLIB) +
DA (&GCUSMKI1 +
&GCUSMNI1) SHR
ALLOC FI (IMSLIB) DA (&GCUSMNI1) SHR
ALLOC FI (SYSLIN) DA (&M5PUNCH) SHR
CALL 'SYS1.LINKLIB (HEWL)' 'LET,LIST,XREF,RENT,REUSE'
FREE FI (SYSPRINT SYSLMOD SYSUT1 SYSLIB IMSLIB SYSLIN)
)ENDSEL
END
```

---

# Index

## A

ACCEPTing APARs, 5-3  
ACCEPTing PTF SYSMODS to distribution libraries, 4-6  
ACCEPTing SYSMODS to distribution librariesdistribution libraries  
    ACCEPTing SYSMODS to, 4-6  
Acrobat Reader, 1-9  
    using, 1-10  
Adobe Acrobat Reader, 1-9  
allocating SMP/E data sets, 4-2  
APAR component identifier, 5-2, 6-1  
APAR modification number identifier, 5-2, 6-1  
APARs  
    ACCEPTing, 5-3  
    APPLYing, 5-3  
    identifying previous RSMs, 5-1  
    RESTORing, 5-3  
APPLY customizing APARs, 5-1  
APPLYing APARs, 5-3  
APPLYing PTF SYSMODS to target libraries, 4-4  
APPLYing SYSMODS to target libraries, 4-4  
APR SYSMODS  
    RECEIVing in global zone and SMP/E, 4-4  
attach facilities, 5-8  
automatic date validation, B-18

## B

banner page  
    displaying, 5-1  
    example, 5-2  
batch query language, 5-13  
BIND function, 5-9  
BLASM#1, 5-6  
BLCOPY1, 2-2, A-3  
BLCOPY2, 2-2, A-4  
BLSMPE#1, A-13  
BLSMPE#2, A-17  
BLSMPE#3, A-22  
BLSMPE#4, A-23  
BLSMPE#5, A-24  
BLSMPE#6, A-24  
BLSMPE#7, A-25  
BLSMPE#8, A-27  
BLSMPE#9, A-28  
BLSMPE#A, 5-3, A-8  
BLSMPE#B, 5-3, A-8  
BLSMPE#C, 5-3, A-9  
BLSMPE#D, A-10  
BLSMPE#E, A-10  
BLSMPE#F, A-11  
BLSMPE#G, A-12  
BLSMPE#H, A-13

---

BLXASM#1, A-29  
BLXASM#2, A-30  
BLXBAN#1, A-31  
BLXCBQ#1, A-32  
BLXCBQ#2, A-33  
BLXCBQ#3, A-33  
BLXCOP#1, A-35, A-36  
BLXDB2#1, A-39  
BLXDB2#2, A-41  
BLXDB2#T, A-38  
BLXDBQ#1, A-37  
BLXDBQ#2, A-37  
BLXINQ#1, A-42  
BLXMSG#1, 5-15, A-43  
BLXOLX#1, 5-14, A-44  
BLXOLX#2, 5-14, A-45  
BLXPAL#1, A-46  
BLXRLK#1, 5-12, A-47  
BLXRSQ#1, A-48  
BLXRSQ#2, A-49  
BLXRSQ#3, A-49  
books, 1-9  
BQL install, 5-13  
BQLPARM, 5-13, B-30

## C

---

CALL attach, 5-8  
CDROM contents, 1-9  
CLIST library, 5-17  
COBOL Quick Start, 5-30  
COMLIB  
    parameters, B-21  
component identifier for PTFs and APARs, 5-2, 6-1  
Computer Associates  
    Total License Care (TLC), 1-10

contacting, 1-10  
    Total License Care (TLC), 1-10  
contacting Computer Associates, web page, 1-11  
CSI, defining zones, 4-3  
customizing, 1-9, 5-1  
customizing APARs  
    APPLYing, 5-1

## D

---

data validation symbols, B-17  
DB2 installation, 5-6  
DB2 Quick Start, 5-30  
default directory, 1-10  
defining SMP/E CSI and zones, 4-1  
defining zones in CSI, 4-3  
displaying banner page, 5-1  
distribution libraries  
    ACCEPTing PTF SYSMODS to, 4-6  
documentation, 1-9  
    installing online books, 1-9  
    viewing, 1-10

## F

---

file tailoring output, 5-20  
file tailoring skeletons, 5-18, 5-29

## G

---

Generate facility, 5-18  
global zone  
    RECEIVING MCS and SYSMODS in, 4-3  
    RECEIVING PTF and APAR SYSMODS in, 4-4

## I

---

identifying previous APARs/RSMs, 5-1  
IMPORT option, 5-18  
IMS attach, 5-8

---

## installation

- BQL, 5-13
- copy files, 2-1
- DB2, 5-6
- JCL, A-1
- M4PARAMS, B-1
- MARKSQL, 5-6
- OLX, 5-14
- OQL, 5-13
- overview, 1-4
- own code, 5-12
- program analyzer, 5-11
- quick start, 5-30
- system tape unload, 2-1
- transfer files, 2-2
- TSO help, 5-14
- VISION:Workbench for DOS, 5-15
- VISION:Workbench for ISPF, 5-16

## Installation Preparation Dialog

- Initialization Display, 3-5
- JCL Part 1 panel, 3-19
- JCL Part 2 panel, 3-20
- JCL Part 3 panel, 3-20
- JCL Part 4 panel, 3-21
- JCL Part 5 panel, 3-22
- navigation, 3-2
- Panel Display, 3-9
- SMP/E structure, 3-4
- Variables Part 1 panel, 3-10
- Variables Part 2 panel, 3-12
- Variables Part 3 panel, 3-14
- Variables Part 4 panel, 3-17

- installation preparation dialog, 1-7, 3-1
- tips, 3-2

- installation verification procedure, 4-5

- installing, 1-9

- Acrobat Reader, 1-9
  - documentation (online books), 1-9

- ISPF, 5-19

- ISPF LIBDEF, 5-19

- ISPF/PDF facility, 5-22

- ISPF, 5-18

- ISPF, 5-19

- ISPLLIB, 5-16, 5-17

- ISPMMLIB, 5-17

- ISPPREP, 5-27

- ISPSLIB, 5-18

---

## J

- JCL Part 1 panel, 3-19

- JCL Part 2 panel, 3-20

- JCL Part 3 panel, 3-20

- JCL Part 4 panel, 3-21

- JCL Part 5 panel, 3-22

---

## L

- LIBDEF, 5-19

- Library Management Facility, 5-30

- library requirements, 5-16

- License Management Program (LMP), 1-2

- licensing, 1-10

- licensing (international), 1-10

- licensing (U. S.), 1-10

- licensing requirements, 1-4

- LMF, 5-30

- load library, 5-19

- locate file tailoring skeletons, 5-18

- locate text, 5-17

---

## M

- M4LEPARM, 5-5, B-16

- M4PARAMS, 5-5, B-1, B-6

- M4SFPARM, 5-5, B-17, B-19

- M9BGTS, E-2

- M9BGUPNL, E-1

- M9FGTS, E-6

---

M9FGUPNL, E-5  
M9FTOUT, 5-20  
M9GCTPU1, E-16  
M9GCTPU2, E-12  
M9GCTSBG, E-12  
M9GCTSFG, E-16  
M9PRIM, D-2  
maintenance, 6-2  
MARKDB2, 5-8  
MARKDLI, 5-8  
MARKIV, 5-8  
MARKLIBP, 5-5, B-21, B-22  
MARKSQL, 5-6, B-23  
MARKSQLC, 5-8  
MARKSQLI, 5-8  
MARKSQLT, 5-8  
MCS  
    RECEIVing in global zone and SMP/E, 4-3  
modification number identifier for PTFs and  
APARs, 5-2, 6-1  
modify parameter modules, 5-5  
MOSAIC processing, 5-7

## N

---

navigating the Installation Preparation Dialog, 3-2

## O

---

OLX install, 5-14  
Online Query Language, 5-13  
Online Query Language parameters, B-37  
OQL install, 5-13  
OQLPARM, 5-13, B-37  
OS/390 SMP/E facility, 1-2  
own code integration, 5-12

## P

---

panel identification, 6-4  
panel library, 5-17  
panel preprocessing, 5-27  
parameter customization, 5-5  
parameters, modify, 5-5  
PDF (Portable Document Format), 1-9  
Portable Document Format (PDF), 1-9  
product licensing, 1-10  
program analyzer install, 5-11  
PTF component identifier, 5-2, 6-1  
PTF modification number identifier, 5-2, 6-1  
PTF SYSMODS  
    ACCEPTing to distribution libraries, 4-6  
    APPLYing to target libraries, 4-4  
    RECEIVing in global zone and SMP/E, 4-4

## Q

---

query language  
    parameters, B-30  
quick start, 5-30

## R

---

RECEIVing MCS and SYSMODS in global zone and  
SMP/E, 4-3  
RECEIVing PTF and APAR SYSMODS in global  
zone and SMP/E, 4-4  
reentrant, 5-19  
RESTORing APARs, 5-3

## S

---

setups, 1-9, 5-1  
site ID, 1-10  
SMP/E  
    APPLY customizing APARs, 5-1  
    RECEIVing MCS and SYSMODS in, 4-3

---

- RECEIVing PTF and APAR SYSMOD in, 4-4
- SMP/E CSI and zones, defining, 4-1
- SMP/E data sets, allocating, 4-2
- SMP/E facility, 1-2
- SMP/E setup, 1-8
- SMP/E structure, 3-4
- startup CLIST, 5-18
- static integration facility, 5-12
- STEPLIB allocation, 5-19
- support, 6-3
- SYSMODS
  - ACCEPTing to distribution libraries, 4-6
  - APPLYing to target libraries, 4-4
  - RECEIVing in global zone and SMP/E, 4-3
- SYSPROC, 5-16
- system link library allocations, 5-19
- system tape, 1-3
- system tape unload, 1-7

## T

---

- target libraries
  - APPLYing PTF SYSMODS to, 4-4
  - APPLYing SYSMODS to, 4-4
- task library, 5-17
- technical support, contacting Computer Associates, 1-11
- Teradata Database System, 5-11
- TLC (Total License Care), 1-10
- Total License Care (TLC), 1-10
- transfer files, 2-1–2-2
- TSO attach, 5-8
- TSO command processor, 5-13
- TSO help, 5-14

## U

---

- unexpected error panel, 6-4

- unload system tape, 2-1
- user code, 6-5
- user panels, 5-29
- using
  - Acrobat Reader, 1-10

## V

---

- Variables Part 1 panel, 3-10
- Variables Part 2 panel, 3-12
- Variables Part 3 panel, 3-14
- Variables Part 4 panel, 3-17
- verifying the installation, 4-5
- viewing documentation, 1-10
- VISION:Workbench for DOS, 5-15
- VISION:Workbench for ISPF, 5-16
  - allocation requirements, 5-16
  - invocation, D-1

## W

---

- web page
  - Computer Associates, 1-11

## X

---

- XSR@PRIM, D-1

